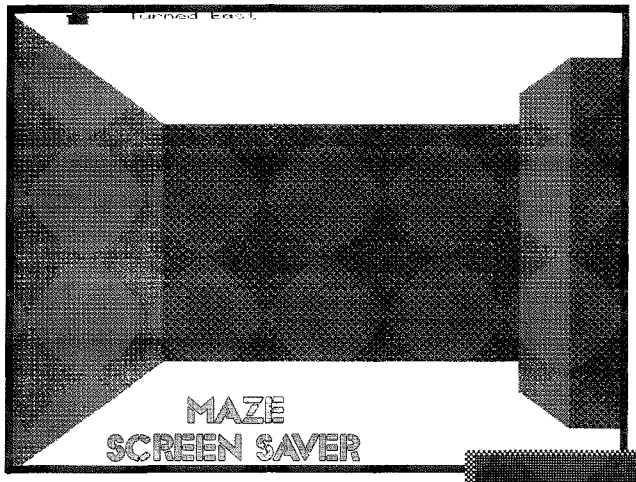
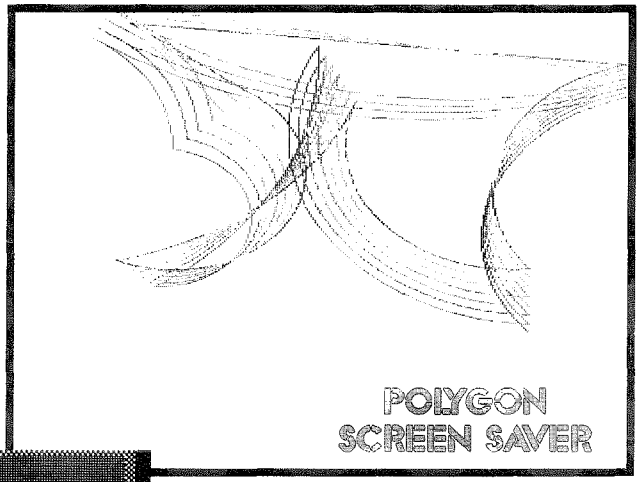
QL Today

Volume 4
Issue 4
Nov/December
1999

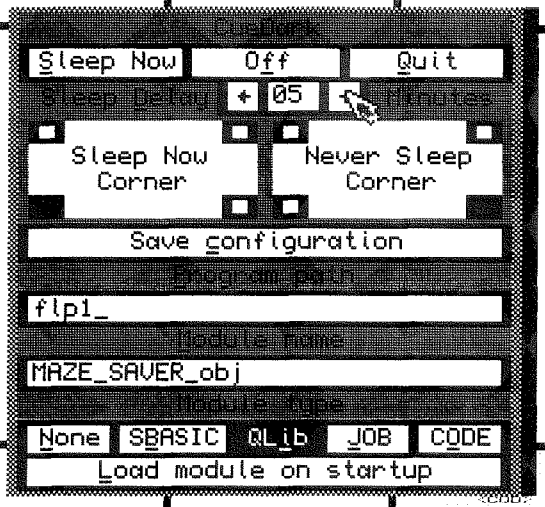
ISSN 1432-5454

The Magazine about QL, QDOS, Sinclair Computers, SMSQ...

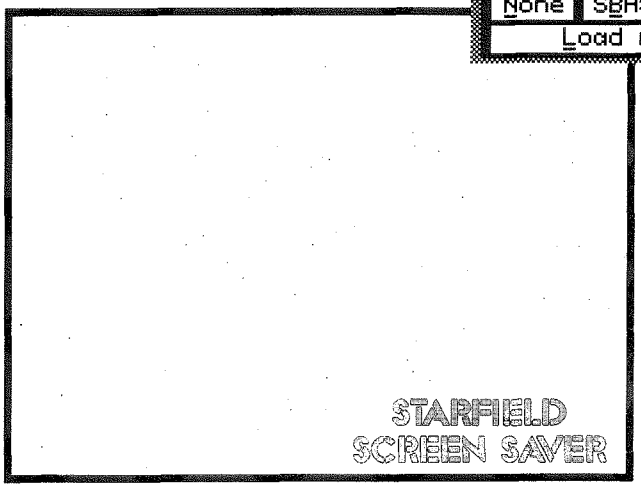


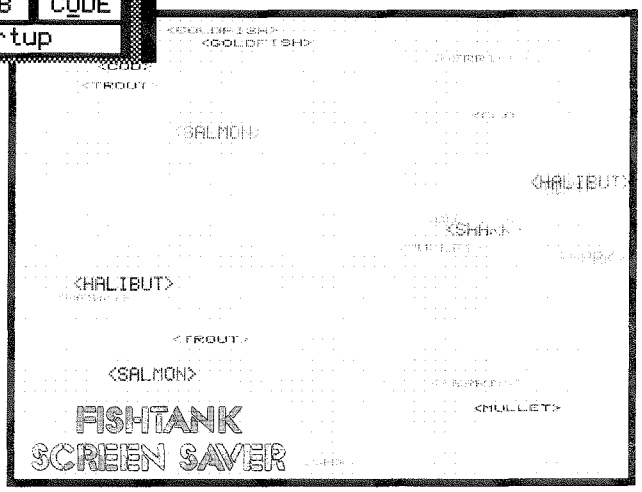


COVER DISK WITH



SCREEN SAVER SPECIAL





PLUS YEAR 2000 QL TODAY CALENDAR

Contents

Advertisers

in alphabetical order

QL Today

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software
Im stillen Winkel 12
47169 Duisburg
Germany

Tel. +49 203 502011
Fax +49 203 502012
Box1 +49 203 502013
Box2 +49 203 502014
email: JMerz@j-m-s.com

English office:

Q Branch
The Bank Volt
6 Coronation Buildings
Ham Road
Sussex BN11 2NN
United Kingdom

Tel. +44 1903 200212
Mobile +44 836 745501
Fax +44 1903 208070
Fax +44 1273 381577
email: qbranch@qbranch.demon.co.uk

Editor:

Dilwyn Jones
41 Bro Emrys
Tal-Y-Bont, Bangor
Gwynedd
United Kingdom LL57 3YT

Tel. +44 1248 354023
Fax +44 1248 354023
email: dilwyn.jones@dj.softnet.co.uk

QL Today is published bi-monthly, our volume begins on 15th of May. Subscriptions begin with the current issue at the time of sign up. Please contact the German or English office for current subscription rates.
Payment in DM (drawn on a German bank) can be made by either Cheque or Euro-Cheque. Payment in £ (drawn on an English bank) can be made by Cheque. Cheques should be made payable to Jochen Merz Software (German office) or QBranch (English office).

Credit Card holders may subscribe by either calling or sending their Credit Card number and Expiry date. Credit cards will be charged in DM (German office) or in £ (English office).

We welcome your comments, suggestions and articles. YOU make **QL Today** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email or into one of the JMS-BBS's. We prefer ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

Article and Advertising deadlines are as follows:

Issue 1: 30 April
Issue 2: 30 June
Issue 3: 30 August
Issue 4: 30 October
Issue 5: 30 December
Issue 6: 30 February

QL Today reserves the right to publish or not publish any material submitted. Under no circumstances will **QL Today** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL Today**. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

This magazine and all material within is © copyright 1999 Jochen Merz Software unless otherwise stated. Written permission is required from the publisher before the reproduction and distribution of any/all material published herein. All copyrights and trademarks are hereby acknowledged.

Editorial

Dilwyn Jones

As we approach the new millennium I wonder how many of us dare ask ourselves just what sort of future it holds for us all as QLers?

For a long time we worried about "colour drivers" - would they ever appear? (they have, for the Q40)
For a long time we worried about "TCP/IP stacks" and email clients and Web browsers - would they ever appear? This issue and colour drivers are surely the two most important immediate requirements for the future of the QL (the two Jonathans, Hudson and Dent, are busily arranging those, one doing the client software, the other doing the low level stuff)
For a long time we worried would we ever have a worthy QL successor? (first the Aurora and then the Q40 came and impressed)

Now comes the threat to our only large international user group as the committee propose to make it a multi-platform group into the next century. That proposal stirred up heated debate in the QL Users Email Mailing List and presumably among the Quanta membership too. One thing is for sure, when it comes to a vote on this issue, those of you who are members must vote one way or another. It is pointless having a vote from say only 5% of the membership as can happen when votes are taken at Annual General Meetings. Whether you want Quanta to remain purely a QL-related group, or whether you want to be freely able to have PCs and Macs and other computers discussed, please vote. The outcome, I feel, has to be decisive one way or another, otherwise it is meaningless.
QL Today will remain a QL-related magazine either way.

We look forward to the QL2000 bash next year. At long last we have a truly international meeting planned well in advance. The Eindhoven and USA shows have long been international shows to some degree, but QL2000 should be something special. Al Boehm is acting as co-ordinator for those travelling from the USA and has already had many enquiries; wonder if such pre planning is going on in other countries too? I would love to be able to meet all those QLers worldwide I have never met but corresponded with over the years. Just think - potentially hundreds of QL users gathering for the "main event" of the QL's recent history!

There's been some doom and gloom over the QL's future. But there's also a hell of a lot to be smug about too! With this issue we bring you not one free extra, but two! We couldn't let the millennium sneak by without a calendar and cover disk. The calendar has a lot of useful QL information and contact details to keep on your wall, while the cover disk contains a lovely little keyboard and mouse aware screen saver program in CueDark. Enjoy them both.

And happy QLing long into the next millennium!

Dilwyn Jones

Cartoon

NEWS

Austrian QL Website

... has found a new home and is now also in English:
http://altern.org/kuelat/index.html

RWAP Software

RWAP Software has now released v3.24 of QL Genealogist which ensures that two family trees can be joined together (no more are marriages 'torn apart' by adding the parents details of both spouses when they have no children.
Q-Route is now v1.08C which is faster than earlier (pre v1.08) versions and fixes a few long standing bugs (including some I introduced inadvertantly in v1.08).
The updated Epson ESC/P2 drivers for ProWesS have now been released, which include better compression techniques (hence faster printing) and 720dpi modes. I hope to add 1440dpi shortly once I have coaxed the details from Epson as to how to control it. As with earlier versions, these drivers will support all RGB colours allowed in ProWesS, not just the few which are currently supported by LineDesign.
If anyone wants any more details, please let me know. Does anyone know what values to use in LineDesign v2.10 with the standard (none colour replacement drivers) to get different colours??
Please note that my phone number as published in the last issue of QL Today was missing a digit! It is now **01902 836888.**

MicroEMACS - Thierry Godefroy

Just to let you know that an updated MicroEMACS v4.00 release (05/11/99 release) is available at the following URL:
http://qdos.cjb.net/english/download.html
Here are the changes since the last release:
31/10/99 Special release for Timothy Swenson.
- Added support for a "uelocal_rc" command file on startup.
- Bugs corrected in "qmenu-*" functions.
05/11/99 MicroEMACS v4.00 patch level 04:
- Corrected bugs into the update_hilite() function that were messing the display when the hilighted area was panned left out of the window or when the hilighted area was larger than the window width.
- Corrected a bug into update_line() function that caused the last line of a higlighted block to be wrongly hilighted up to the right screen edge of the screen when this last line got trailing spaces.
- Improved the right justification macro in "wpage_cmd".
- Added the DRED (dark red), DGREEN (dark green) and GOLD colours to MicroEMACS for QDOS/SMS. These are in fact patterns (respectively red/black, green/black and red/green)...
- The version number shown into the about box now includes the patch level of MicroEMACS v4.00. This patch level reflects the number of bug fixes or improvements (relevant to all plateforms) I brought to MicroEMACS v4.00, while the QDOS port is still characterized by the release date.
- Added the "grabbing hand" sprite for dragging sub-window events.
- MicroEMACS no more attempts screen operations while its screen is locked (by the pointer environment). Instead it resumes operations and marks the screen as garbage so that a redraw is forced when the screen is unlocked again. This allows MicroEMACS to execute the CSM requests while its screen is locked, provided that the cursor was active (flashing, i.e. the pointer was not showing) before MicroEMACS screen was locked.

FileInfo II - Thierry Godefroy

FileInfo II v3.40 is now released and available from my web site
http://qdos.cjb.net/english/download.html
Here is the changes record:
- When presenting the "Actions" menu, FileInfo II may now remind you on what "file to process" these actions are to be applied (configurable with (Menu)config).
- Additional parameter feature implemented as well as the corresponding changes to the history, FileInfo II extensions, S*BASIC PROCs/FNs and fi2_call() library function (libfi2_a is now v1.20).
- A bug corrected where commands line to be stuffed into the keyboard queue was not sent if the "wait for end of processing job" flag was set (or forced via the bit 31 of the key).

New EPSON Printers

The new range of EPSON Stylus Color printers is now available - they all end with a ..60. The cheap model is the 460, next one up 660, 760 and top model 860. The main difference seems to be the printing speed. Disappointing: they are all ***ESC/P2 raster*** only printers, not full ESC/P2 - and therefore not suitable for QDOS and SMSQ/E. The top-end models 850 and 900 still come with the full ESC/P2 set and can be used with QDOS, and the 740 can also be used. If you want to get a nice printer, best decide now and get yourself a Stylus Color 740 before they are sold out.

QL - PC Transfer Utility

JUST WORDS! has now released its QL - PC word processing file transfer utility. QL-2-PC TRANSFER is a versatile and simple to use pointer driven program covering a wide range of possible transfers.

QL files for transfer can be in Quill, Perfection, Text87 or ASCII format and transfer can be to either a WINDOWS or DOS based word processor. PC files for transfer must be in ASCII format.

In all transfers accented and similar characters are correctly converted to the new format, which makes this program interesting for any QL user wishing to transfer a non-English language text.

Depending on the word processing software it may be possible to transfer bold, underlined and italic text. The program can generate Rich Text Format code for importation into windows based word processors, or a pseudo WordPerfect 4.2 file for loading into DOS based word processors.

Users of a windows based word processor can also set font, font size, justification, margins and tabs on the QL. This formatting information can be saved to disk, a feature users of a standard "house" style will find invaluable.

QL-2-PC TRANSFER also contains experimental routines for extracting the text message from e-mail files, and for tidying up text files by removing extraneous line feeds, spaces and soft hyphens. This generates a text that is easier to re-format in a QL word processor, and is a useful feature for anyone who downloads files from the internet or who uses an OCR-reader.

QL-2-PC TRANSFER costs just £10 or 15 Euros and is available either directly from JUST WORDS! or from QBRANCH.

QBRANCH News

1. Mark Knight has released the Fractal Collection as promised in the last issue. He is also pretty far advanced with The Knight Safe III. The new version will be split into two parts one for backup and one for restoring. This should make it quicker and easier to use. It will also have a new menu which makes it easier to extract a single file from an archive and some other improvements.

2. Mark Swift told me at the Stafford show that the latest version of QDOS Classic for the Q 40 can now be run over the top of SMSQ/E making it possible for users to have access to both systems. QDOS Classic is free and downloadable from Mark's Web Site (I don't have the URL to hand). He has also patched the original JS ROM to remove the 4Mb limit so users of the various emulators which require a ROM image can specify higher memory values. This is not fully tested yet so he would welcome volunteers.

3. A second test version of the colour drivers has arrived for the Q 40 and I am evaluating it now. Still not quite released but so close you almost smell it. A few things still need ironing out but it is looking good. Simon Goodwin took a copy of the specs away with him and said he would try to write some graphics routines for them so the next show should be very interesting.

4. We have some second user Aurora/super-Hermes/Qubides etc. Some parts from the Tower cases mentioned in the last issue have been sold so we have split the units up.

5. Please accept apologies to anyone who turned up at the Eindhoven meeting to see Q Branch. I have been doing a little Rock'n'Roll work this month and they moved the dates forward making it impossible for me to attend.

More RWAP News

Latest news is that I have now re-released FlightDeck v1.04 - this is an excellent Flight Simulator for the QL, based on a twin engined passenger jet, with full 3d shaded views of the outside world. Worlds can be defined by the user, but the one supplied comes with details of 25 major UK airports and over 200 navigation beacons. Cost is £10.

I have now received (many thanks to Chas Dillon), the source code for Cash Trader and Payroll formerly sold by Digital Precision and PDQL.

I am looking at releasing updated versions of these programs to overcome the problems with the year 2000 - it does not seem too difficult.

If anyone is interested in getting updates to these programs, then please contact me. I would also welcome any reports of bugs in these programs (to start with, ideas for updates can be sent at a later date).

Okay, I thought that if you are all so interested, I could let you have the details as to where these source files are stored. According to the text, Chas has arranged with all the copyright holders the ok to distribute the sources as PD.

The address to look at is:

http://www.realcom.co.uk/ql_thor/

There are source codes for TURBO and EDITOR as well as other various programs.

What I would emphasize is that you need to look at the conditions and someone needs to co-ordinate any efforts to improve this software.

I myself am only looking at Cash Trader and Payroll at the moment (I have possibly already fixed the Y2K problem with Payroll). I understand that Mark Knight wants to co-ordinate work on Turbo.

■

More QL-Related Web Sites

http://www.breezer.demon.co.uk/
(John Garner World Of Sinclair page, mostly Spectrum/Z88, some QL links)

http://www.alchemist.clara.net
(Alchemist Research web site, mostly Z88 related)

http://twinpentium.lcp.linst.ac.uk/colhome/colhome.htm
(Dr. Colin Parsons home computers page, including QL)

http://www.pncl.co.uk/~prospero/dhretro.html
(David Harrison retro computing site - links to a number of QL sites)

http://wuarchive.wustl.edu/pub/aminet/info/www/home.html
(QDOS4Amiga Amiga QL emulator available from Aminet site)

Millennium Meeting by Al Boehm

Quanta is thinking about having a special QL Millennium Meeting in the year 2000 and inviting overseas guests. NESQLUG has appointed me, Al Boehm, as point of contact for the meeting. So far 13 people in the US have indicated an interest in such a trip. If you think you might want to go, even if non-NESQLUG, contact me. My email is: **boehm@ziplink.net** and my mailing address is: 2501 Ermine Rd., Huntsville, AL 35810, USA.

You and Your Software - Just good Friends? - Part 5

Problems not solutions (2)

Geoff Wicks

Of the many thousands of words I have written for QL Today, one article is remembered more than any other. Two years ago I wrote a tongue in cheek attack on buttons that divided the QL community. One group of users would have had me shot at dawn, while another would have erected a statue in my honour. Fortunately the third and largest group said it had made them think about their own button use.

Readers of QL Today, who are also members of QUANTA, will know that recently a similar debate has been raging in the columns of the QUANTA magazine. There is probably no subject that raises the QL passions more than the merits and demerits of the pointer environment.

As a software author you have to take a position in this debate. You may be able to get away with producing both pointer and non-pointer versions of a program, but this will involve you in extra work, and you may find this restricting your style.

Just Words! moved from entirely non-pointer to predominantly pointer programs, because this was the most common single request I had from customers. People are now using high resolution screens, and many are using systems like QPC where their hardware includes a mouse or similar device. The only compatible operating system for this hardware is SMSQ-E with the pointer environment built in. Nevertheless we should never forget that there are still many QL users who do not use the pointer environment. There are even expert programmers, who have contributed much to the QL, who abhor it. But just as most PC programs are now windows compatible, QL programs will increasingly become

Im stillen Winkel 12 D-47169 Duisburg
Tel. 0203 502011 Fax 0203 502012
http://www.j-m-s.com/smsq/index.htm

Jochen Merz Software Applications and Version Numbers

QPC1 + QPC2 + SMSQ/E Software QL-Emulator for PC's	[V1.52]	**DM 249,-**
QPC1 + QPC2 + SMSQ/E for SMSQ/E owners	[V1.52]	**DM 199,-**
QPC2 Upgrade from QPC1	[V1.52]	**DM 79,90**
SMSQ/E for ATARI or QXL or (Super)GoldCard	[V2.93]	**DM 249,-**
QD 98 Pointer-Environment-Editor	[VA.06]	**DM 119,-**
QD 98 Upgrade from V9	[VA.06]	**DM 39,90**
QD 98 Upgrade from previous versions	[VA.06]	**DM 49,90**
NEW! AGENDA for WMAN or ProWesS	[V1.02]	**DM 59,90**
QMAKE Pointer-driven MAKE for GST/Quanta Assembler	[V4.25]	**DM 45,-**
BASIC Linker	[V1.12]	**DM 49,-**
WINED Floppy/Harddisk Sector- & File-Editor	[V1.21]	**DM 49,-**
FiFi II File-Finder - Extremely useful!	[V4.19]	**DM 49,-**
FiFi II Upgrade from earlier versions	[V4.19]	**DM 19,-**
EPROM Manager	[V3.02]	**DM 49,-**
QSpread99 Spreadsheet Program	[V2.05]	**DM 139,-**
QSpread99 Upgrade	[V2.05]	**DM 39,90**
QPAC I Utility programs	[V1.07]	**DM 59,-**
QPAC II Files, Jobs & other Things	[V1.39]	**DM 119,-**
QPAC II Update	[V1.39]	**DM 16,-**
QTYP II Spell checker	[V2.17]	**DM 79,-**
QMENU The Menu Extension Programming Manual	[V7.04]	**DM 41,90**
QMENU Upgrade	[V7.04]	**DM 18,-**
QPTR Pointer Toolkit	[V0.30]	**DM 89,-**
QPTR Update	[V0.30]	**DM 16,-**
DISA Interactive Disassembler	[V3.04]	**DM 89,-**
DISA Upgrade from previous versions	[V3.04]	**DM 33,-**
typeset-ESC/P2 text87 driver for all ESC/P2 printers (incl. Stylus)		**DM 65,-**
CueShell	[V2.14]	**DM 90,-**
CueShell for QPC	[V2.14]	**DM 40,-**

Even if you already own QPC, you can upgrade to include CueShell for DM 40,-!!!

SER Mouse software mouse driver for serial mice		**DM 25,-**
EASYPTR Part 1&2	[V3.10]	**DM 89,-**
QDOS/SMSQ Reference Manual		**DM 79,-**

TERMS OF PAYMENT

Postage and package **[Germany]** DM 8,99 (if total value of goods is up to DM 50,- then only DM 5,99). **[Europe]** DM 14,50 (if total value of goods is up to DM 50,– then only DM 9,50). **[Overseas]** between DM 14,50 (1 item) and DM 35,– (maximum). All prices incl. 15% V.A.T. (can be deducted for orders from non–EEC–countries). E&OE. Cheques in DM, £'s, Eurocheques and Credit Cards accepted.

EUROCARD VISA MasterCard

pointer compatible. Do not be put off by the supposed difficulties of writing pointer programs. If you are a reasonably competent Basic programmer, you have the abilities to write a pointer program using Albin Hessler's Easyptr suite. Once you have mastered your first pointer program, the second and third come easily. The difficulty with Easyptr is the manual not the programming.

Another change in Just Words! programs was a move from Turbo compiled to QLiberator compiled programs. Pointer programs cannot be compiled by Turbo, a situation that could change in the future. Turbo and the Turbo Toolkit versus QLiberator is another subject that raises passions in QL users. Indeed some of my customers expressed their distaste that they had to "dirty" their machines to run my programs as they used commands in the Turbo Toolkit.

The battles between Turbo and QLiberator are not just a matter of history, but are still present today. QLiberator gained the upper hand because of its compatibility with the pointer environment and modern hardware, but the detractors of Turbo and Turbo Toolkit should never forget they kept the QL alive when it was a much slower machine. The Turbo Toolkit provided the first, and in my opinion still the best, error trapping usable on all ROMs. It also had commands like PEEK$ and POKE$ which years later were finally implemented in SMSQ. A software team are looking at Turbo and it could gain a new lease of life with possible pointer compatibility, again rivalling QLiberator.

When Just Words! changed to QLiberated programs I had to write routines to replace PEEK$, and these slowed the programs down. Just Words! programs run slower than when they were Turbo-compiled, but few users have noticed this because of some programming tricks.

A good programmer knows how to fool the user that his program is working faster than it is. The golden rule is never to leave a screen blank while your program is doing its work. Always give the user something to look at, even if it is only a row of dots being printed to the screen. If nothing else he will know his machine has not crashed.

Some users have expressed surprise at how quickly my Style-Check program analyses a sentence, but this is an illusion. The analysis is a slow process that is being done and stored while the sentence is printing to the screen. This slows the printing of the sentence, but few people notice it as the printing is continuous. What they notice is the instant printing of the stored analysis when the sentence is complete.

One of the reasons Just Words! will probably continue to QLiberate its programs is that it is possible to build machine code routines into a program using the $$asmb command. This is something that is not possible with Turbo.

This brings us to the whole problem of resident extensions or the machine code routines that have to be loaded before a program will work. Most of us use several of these in our day to day QL work. We implement toolkit 2; we load extensions to run the pointer environment; and we load some or all of QTYP, QMENU, the Turbo Toolkit, Simon Goodwin's DIY Toolkit or the Qliberator runtime code. These extensions allow us to do clever things with our machines. They provide us with new commands, but they have their disadvantages. Extensions should always be loaded in the boot program at the start of a computing session, otherwise you could have memory handling problems. Once loaded, extensions cannot usually be removed. The more extensions you use, the greater the chance they will interfere with one another.

If you have the programming ability to write extensions, should you use them or should you build the machine code routines into your program? Unless you are a top programmer, I would choose the latter. The first versions of Solvit-PLus used a small machine code extension. It was only just over 1K in length. If a user uses the program for a couple of minutes to check a crossword, that 1K remains unused in his machine during the rest of the computer session. The routine is now built into the program and in thus only loaded when the program itself is loaded, and removed when the user quits the program.

QL users tend to be devoted to the extensions they load in their boot programs, and they expect your software to conform to the extensions they load. Again this is a no-win area for the programmer, and you have to find your own level of compromise. Do you make your programs QMENU compatible or not? Do you incorporate the runtime extensions in QLiberated programs? Do you use the Turbo Toolkit? What level of downward compatibility are you prepared to cater for? We can assume that most QL users now have disk drives and use Toolkit 2, but at what stage can we say that SMSQ-E is the norm and QDOS compatibility is no longer important?

There will always be problems without solutions.

Next Time: Correcting the errors of your ways.

The CueDark Screen Saver System

We have a real treat for you with this issue. Albin Hessler, author of CueShell and EasyPtr, has kindly allowed us to distribute his CueDark screen saver system on a QL Today cover disk.

CueDark works with QDOS or SMSQ systems as long as pointer environment is present, and is both mouse and keyboard aware, unlike some older QL screen saver systems.

The basic principle of a screen saver is that if you do not use the computer for a given time period, a program cuts in to blank the display to prevent screen burn-in, or simply to hide the screen display from casual eyes if you have confidential material on screen and leave your desk for a few minutes. The normal display is restored when you press a key on the keyboard, or move the mouse.

While CueDark can simply blank the display like other screen savers, it can also run program modules to provide other attractive displays. Several examples are supplied, and the necessary programing information is supplied to allow you to write your own modules in assembler, SBASIC or QLiberated SuperBASIC for example.

CueDark can run modules written in SBASIC for SMSQ or SMSQ/E users. Or you can write your own assembler programs and run them as Jobs or as modules specifically written for CueDark using the programming information supplied.

CueDark will even supply display size and program path information to your program, to allow you to write saver programs which will work on all display resolutions to save you some hard programming graft!

Setup instructions are shown below. Other information (including programming information) is in the file called CUEDARK_DOC (a Quill DOC file) on the cover disk. If you prefer, there is also a plain text file version called CUEDARK_TXT on the disk.

Please make a backup copy of the cover disk and read the instructions carefully before you attempt to use the program. It will run directly from the cover disk if you insist, but we urge you to make a backup copy in case something goes wrong and the disk gets damaged!

The copy on the supplied cover disk is configured to run from FLP1_. Anywhere else, and you'll have to reconfigure it.

In order to use CueDark, you have to be using pointer environment. A mouse is helpful, but not essential. Toolkit 2 is also preferrred, especially if you plan on writing your own screen saver modules. Nowadays, Toolkit 2 (by Tony Tebby) is one of those essentials present on just about every system anyway.

CueDark is Copyright (C) Albin Hessler.

Albin asks that where possible any modules written for the CueDark system are made freely available for the benefit of all CueDark users. To that end, we intend to run a competition to find the best CueDark module written by a reader and we hope to include the best of those received on our next cover disk.

FEATURES LIST

* Mouse and keyboard aware screen saver system
* Simple to use
* Compact controller program runs as a normal job
* Variable delay time
* Several example screen saver routines supplied
* Write your own alternative screen saver routines
* Requires pointer environment
* "Parking" areas for pointer to force or disable savers
* Fully configurable, from within CueDark itself!

CueDark Setup

1. If you have a hard disk, create a directory to hold the program, e.g.

```
MAKE_DIR WIN1_CueDark_
```

2. Copy all the files from the cover disk to where they will be kept, e.g.

```
WCOPY FLP1_ TO WIN1_CueDark_
```

3. To configure the program, execute CueDark itself:

```
EXEC WIN1_CueDark_CueDark
```

See also the README_DOC file on the disk for latest details on configuration due to last minute changes to the program.

4. Park the pointer at the bottom left of the screen to force the module start screen to appear. Within 10 seconds, while the CueDark text is shown, press the F3 key. The configuration screen appears. Of the 3 options at the top, "Quit" removes the program completely from memory.

5. Set the sleep delay in minutes - the time of inactivity before the screen saver appears. A reasonable value for this is 5 or 10 minutes. You can obviously set it again later if you decide the delay is not suitable for you. You can either hit on

the arrows either side of the delay time display box to increment the value, or click in the box and type in the number of minutes directly.

6. Next click on the small squares in the corners of the diagrams below to indicate which corner the mouse pointer needs to be parked in to immediately start a screen saver (this feature is useful if you have something on screen you'd prefer casual eyes not to see while you leave your desk for a few minutes) and which corner the pointer can be left in to prevent the screen saver appearing at all (useful if you have an advertising display running on your computer which must not be interrupted). By default, if you hold the pointer in the bottom left of the screen, this forces the screen saver to start, and if you hold the pointer in the bottom right of the screen, it prevents any screen saver from starting at all.

7. If the path name shown in the box under "Program Path" is not the one you want (the directory you created above, the location where the program is to run from), click in the box or press P and edit it to Win1_CueDark_ for our example above. The screen saver modules will also reside here. CueDark also passes this path name information to the modules it starts, so they know where they started from and can load any additional files they require from the same location. This saves you the programming effort of adding configuration blocks to your programs, especially when using SBASIC applications.

FAULTY DISKS

If you suspect your cover disk is faulty (e.g. it gives a Bad Or Changed Medium error when you try to copy it), please contact Jochen Merz Software or QBranch to obtain a replacement. The addresses are inside the front cover of QL Today.

8. Next, you need to specify the module type at the bottom of the window.
"None" means don't start any module, just use the in-built screen blanker instead.
"SBASIC" means Start an SBASIC program - you need to be running SMSQ (QXL) or SMSQ/E (which includes QPC) for this option to work.
"JOB" means start a screen saver you can execute as a standard job, e.g. a machine code program you have written following Albin Hessler's programming notes.
"CODE" means an assembler module specifically written to the CueDark specification, see instructions on the disk.
"QLIB" means a QLiberator compiled BASIC program.
If you select an option other than "NONE" the options under and above these options become available for specifying the module name and actions.
Under "MODULE NAME" is a box into which you can type the filename of the screen saver program CueDark is to execute. This should NOT include the directory or "path" name. For example, if the saver program is called MySaver_EXE and saved in the WIN1_CueDark_ directory (total filename WIN1_CueDark_MySaver_EXE) you should only enter MySaver_EXE here. Note that for SBASIC files, you should enter the _BAS or _SAV extensions, even though SBASIC normally allows you to omit these. "Load Module On Startup" at the bottom of the setup box loads a module into memory as CueDark starts, rather than executing it each time from disk as it is required. This would be useful for floppy disk users as they would not have to ensure that the saver disk is in the drive each time it is likely to be needed. It does not work for SBASIC programs stored in the _SAV file format (_BAS should be OK), though.

9. Now click on the "Save Configuration" item, and a copy of CueDark will be saved with the filename made up of the pathname and "CueDark" (WIN1_CueDark_CueDark in our example above)

10. Click on whichever of the three options you require at the top of the screen - QUIT removes the program from memory completely. Note that the module name is not immediately updated - the old name will still be shown as the module CueDark is starting. So it is best to press a key or move the mouse to prevent the 'old' module being started, then the next time CueDark starts a module, it will start the new one you have just specified during configuration.

How to use CueDark

CueDark is a simple executable program. So simply start it with an EXEC command, near the end of your boot program if you wish:

```
EXEC WIN1_CueDark_CueDark
```

That's all there is to it. Once you have got used to the Setup options, you can sit down with the programming notes and start to create your own colourful, weird and wonderful screen saver modules to add to those we've supplied on the disk. Remember that the basic aim is to prevent static text or graphical displays from burning onto the

screen display. Keep it moving, bright and colourful! I expect most users will start by writing their own modules in SBASIC, Compiled BASIC, or copy the example assembler files Albin has supplied. CueDark supplies information to the saver module about the screen resolution and path from which the saver was executed. This information is supplied as a command string and may be extracted using the CMD$ variable in SBASIC for example. I suggest you place a few lines of BASIC like this at the start of your program:

```
100 passed$ = CMD$
110 xpixels = HEX(passed$(1 TO 4))
120 ypixels = HEX(passed$(5 TO 8))
130 path$ = passed$(10 TO LEN(passed$))
```

SBASIC allows line 130 to be abbreviated to

```
130 path$ = passed$(10 TO)
```

The variable xpixels now contains the screen width in pixels and ypixels contains the screen height in pixels. This will help you to scale your graphical and text output to cover the entire screen if you wish.

The variable path$ contains the drive and directory name from which CueDark executed the module. This may be ignored if you wish, or if your module needs to load additional files, this gives your program the means to determine where to load them from without having to program any additional configuration items, as long as your module and any related files can all live in the same directory.

Here is a very simple example screen saver module which simply makes the screen all black, green, red or white in turn.

```
100 passed$ = CMD$
110 xpixels = HEX(passed$(1 TO 4))
120 ypixels = HEX(passed$(5 TO 8))
130 path$ = passed$(10 TO LEN(passed$))
140 WINDOW #0,xpixels,ypixels,0,0
150 REPeat loop
160  FOR paper_colour = 0,2,4,7
170  PAPER #0,paper_colour : CLS #0
180  PAUSE 50*RND(10 TO 60)
190  END FOR paper_colour
200 END REPeat loop
```

It really can be that easy to create your own colourful screen savers which are limited only by your imagination, thanks to the combination of CueDark and SBASIC on the QL! For me, this is ample reason in itself to be using SMSQ in one of its guises for a QL or compatible computer rather than sticking to vanilla QDOS.

Several SBASIC example screen saver modules are on the cover disk, they all have filenames which end with _BAS. Although it is not currently possible to run these as QLiberator compiled jobs, I have tried to write them such if/once it becomes possible to run them compiled (e.g. for QDOS systems) it should be possible to use them with little or no changes.

Please note that there may be other listings from the magazine on the cover disk as well - please be careful not to attempt to use these as screen saver modules, a full list is shown elsewhere.

COMPETITION

We hope you enjoy using CueDark and find it useful.

To encourage development of new modules for CueDark, we will be holding a competition over the next few months to find the best module for CueDark. The best few modules will be placed on our next cover disk (expected to be distributed at the end of this volume of QL Today), and the very best ones will be rewarded with a prize from JMS.

That's right, I have twisted Jochen's arm and persuaded him to part with a copy of CueShell for (a) The best SBASIC or Compiled BASICscreen saver module

(b) The best Job or assembler module (i.e. non-SBASIC/compiled BASIC)

Send your entries to Dilwyn Jones at the usual address shown inside the front cover, along with your name and address of course, and a brief text to indicate what it's intended to do and any special requirements.

Potential Problems

1. At the time of writing, it was not possible to use CueDark on some QDOS systems, in particular we had problems with it on a JM ROM QL and QIMI, and on an Atari ST QL emulator with older QDOS style drivers. If you discover any other incompatibilities, we would be glad to hear from you, or you can report them direct to the author via email at **albin.hessler@t-online.de**

2. CueDark seems to supply screen resolution information based on the screen resolution in effect at the time it was started. So if you alter DISP_SIZE after starting CueDark, it may not be aware of the changed screen size and supply the old information. So if you need to alter display resolution in your boot file, the DISP_SIZE statement should precede the line which launches CueDark.

3. If additional information becomes available at the last minute, we will include this in a file called README_DOC on the cover disk - please read this file if it has been necessary to include it.

A new "SET/ALTER"

George Gwilt

Old SET/ALTER

The DIY Kit procedure SET does not work well with SMSQ/E. In the Master Basic, if you SET a variable, you cannot PRINT it! However, if you put another variable equal to it you can print that. For example:

```
SET aw$ to "Peculiar"
PRINT aw$
```

gives "invalid parameter" on #0

```
a$=aw$:PRINT a$
```

gives "Peculiar" on #1

```
PRINT aw$
```

now gives "Peculiar" on #1

If you enter "NEW", the previously SET values all disappear. If you are in SBASIC, SET anything just gives you "invalid parameter".

New SET/ALTER

The new version of SET works with both the Master Basic and SBASICs. A variable SET in Master Basic can be accessed for use (eg printing and ALTERing) in an SBASIC, but a variable SET in SBASIC although available in that SBASIC is not available in the Master Basic, or any other SBASIC.

"NEW" will leave the SET variables alone - they are still accessible.

The trick is to link in the code by using BP_INIT (vector $110) to set the variables up as functions. Otherwise the code is nearly unchanged. The original code foresaw ALTER being used in compiled programs. The variables SET in the new version are linked in to the currently running program (master Basic or SBASIC). In a program compiled by TURBO, ALTER will work, but SET will not, failing with the message 'bad parameter'.

In both the new and old versions of SET, you can prefix the variable name with "#" so that the presented value of the variable is that SET plus the base of system variables. Whatever name is initially chosen, the value stored is a word. However, in the old version which appeared in QL World May 1991, if you ALTER the SET variable, the new value will only be a word if the name ends with "%". If the name had no final character, the new value would be stored as a float. This would cause the ALTERed variable to give odd results or even crash the QL since the integer would be taken as the mantissa of the floating point number, typically about 2050. The crash would occur if the mantissa were odd.

In the new version, ALTER determines whether it is dealing with a "#" variable (a vector) and if so checks that the result is an even word, or signals an error if not.

Differences in Coding

The listing (can be found on next months cover disk) gives all the assembler instructions, but does not indicate the differences from the original version. There are three alterations which can usefully be described.

1. The use of BP_INIT

At MAKE_VAL the old version put the address of the function being SET directly into the Name Table, as well as making it a function type (9). The new version uses QDOS to link in the function. This is done in MAKE_VAL by filling in the details of the function to be linked in and then calling BP_INIT. The information is held in the first 42 bytes of the user heap. This is initialised the first time SET is encountered to contain three words, 0, 0 and 4. The first two signal "no procedures" and the last, 4, signals one function the length of whose name can be up to 31 bytes. There follows the word offset to the address of the code for the function and then the name of the SET function. This name can't be found by the QDOS CA_GTSTR routine since it was not entered as a string. Instead it has to be picked out of information in the Name Table held at (A6,D7.L). D7 was set at NOTYETSET to the appropriate value. The word at byte two contains the offset of the name from the start of the Name List.

2. The pointer to the BASIC variables

The original SET/ALTER assumed that there was just one Name Table and Name List, held by Job 0, SuperBASIC. This was accessed by finding the base of that program via MT_JINF.

Under SMSQE SuperBASIC has two sets of Name Tables and Name Lists. One is accessed by SuperBASIC's A6 and the other in two ways, by using MT_JINF to get SuperBASIC's base, or by taking the address stored at SV_BASIC plus $68. The information required for SET/ALTER appears in the set accessed by A6 and not in the other one.

When the program using SET or ALTER is not compiled it must be running either under SuperBASIC or an SBASIC. In either case A6 can be used to point to the appropriate base for Name Table or Name List. However, if the program is compiled, its A6 will be inappropriate. In that case access to the Name Table or Name List must be made via SuperBASIC's A6. Thus code following the label "SET" puts A4 equal to the current program's A6 if not compiled, and equal to SuperBASIC's A6 otherwise.

Professional & Graphical Software

ProWesS

ProWesS is a new user environment for the QL. ProWesS is short for "PROGS Window Manager", but it is much more than that. Apart from a new window manager, it contains all the system extensions from PROGS, and is essential if you want to run programs which need these extensions.

The ProWesS reader is a major part of the package. It is a hypertext document browser. This means that text files which include formatting commands (including pictures) and possibly links to other files can be displayed and read in this program. This is used in ProWesS to read (and possibly print) the manuals, and display the help files. The hypertext documents which are used by the ProWesS reader are in HTML format, the format which is popular on Internet to display World Wide Web pages.

Another important aspect of ProWesS is the possibility to allow programs to automatically install themselves on your system, and to be able to run them without resetting the system. This means that, when you get a new program, all you have to do is insert the disk and indicate "start the program in flp1_", a menu option in the "utilities" button. To install a program, you indicate "install software", and the software can be added to your system. This way, you don't need to know how to write a boot file to use the multi-tasking capabilities of your computer.

ProWesS includes many programming libraries. These include syslib, an interface to the operating system, PROforma, a vector graphics system, allowing rendering both on screen and on paper (via a printer driver). The DATAdesign engine is also part of ProWesS. It is a relational database system with a bonus, as you don't even need a key field. You get a powerful record at a time data manipulation extension to the language you already use. Of course it also includes ProWesS itself, the new resolution independent window manager.

PWfile

PWfile is a file management program. It allows you to do all kinds of manipulations on files, like copy, move, rename etc. This allows you to manage your files properly.

To make everything easier, you can choose whether the actions also have to apply to the subdirectories. Also, the program will display how much space the indicated files take up, so that you can know in advance whether there is enough room on the destination device. There are even facilities for copying to DOS disks. You can always limit all selections to file in or excluding certain extensions etc.

PFlist

Easy to use program to create listings on any printer (especially inkjet and laser). This ProWesS application allows you to indicate the files which have to be printed. Each column contains a footer which can include the filename and filedate. The listings always allow perforation. PFlist can create your listings in two columns and in landscape (or both).

fsearch

File search utility with many useful options, like the choice to search only files with a certain extension, and whether or not the directory tree has to be scanned. All occurences of the searchstring will be displayed with line number or offset. You can also use special matching features, like case dependent, matching a space with a stretch of whitespace, and searching for a word dilimited string.

font-utils

manage your font collection. You can preview fonts on screen, see what characters exist in a font and convert Adobe Type 1 and similar fonts for use in ProWesS.

LINEdesign

Create artistic drawings, technical drawings, process bitmaps (even scale and rotate them!), and any kind of vector drawings. You can use grpahics objects to create the most fabulous drawings ever seen. Because LINEdesign is a vector drawing program, any part of the picture can be moved, scaled, rotated, slanted without any loss of precision or resolution. In LINEdesign, pictures are device independant, meaning that the printout will be the same on any printer (e.g. same size and position).

LINEdesign is good at handling text. You can easily put titles and full paragraphs on the page. All the fonts can be displayed at any size, rotation, etc. All the fonts which are available to ProWesS can be used in LINEdesign.

LINEdesign is a drawing program, but it can also be used by people who are not good at drawing. LINEdesign is a great program for making leaflets, posters, and any kind of printed work. Lots of clipart and extra fonts are available from public domain libraries and BBS's. You can even import Adobe Illustrator files.

DATAdesign

Never before has it been so easy to create, fill in and maintain your personal databases. To start a new file, just type the names of the fields. To add or delete a field, no problem, just do it. To change the name of a field, just indicate it. You can choose which fields are displayed and also which records. You can have a hidden comment for each record, look at the file in tabulated form and transfer data to the scrap or hotkey buffer. Files can be memory based (for speed) or disk based (for safety).

Dr. Fr. Hemerijckxlaan 13 /1
2650 Edegem
Belgium

www : http://www.triathlon98.com/PROGS/
email : PROGS@triathlon98.com
tel : +32 (0)3/ 457 84 88 fax : +32 (0)3/ 458 62 07

ProWesS - BEF 2400 DATAdesign - BEF 1200 PWfile - BEF 900 PFlist - BEF 600
LINEdesign - BEF 1200 fontutils - BEF 1200 fsearch - BEF 600

Payment terms :

You have to run ProWesS to make LINEdesign, DATAdesign, fsearch, fontutils and PFlist work (even though DATAdesign uses wman).

All our software is normally supplied on high density (HD) disks. However they can be obtained on double density (DD) disks at an extra costs of BEF 100. To use ProWesS and any of our other packages, you need a system with at least 2MB of memory. You should have a harddisk although a two disk system will also work. The use of SMSQ/E is strongly recommended for optimal use of ProWesS.

If you are VAT registered (specify registration number) or live outside the EEC, the amount to be paid is the total (including postage) divided by 1.21 (no need to pay too much).

Payment can be done by EuroCheque in BEF, or by VISA, EuroCard or MasterCard. Credit card orders can be handled by phone. For credit card, please specify name of card owner, card number and expiry date.

Postage : Costs of postage and packaging have to be added. You can choose the quality. Rate depends on no of programs.

	priority mail			*ordinary mail*		
copies	Belgium	Europe	World	Belgium	Europe	World
one	100	200	240	100	120	145
two	135	340	420	135	190	230
3 or 4	160	560	770	160	310	395
5 to 8	185	870	1250	185	550	705
more	295	1130	1610	295	800	1030

All prices are in BEF, including 21% VAT

Note that the test of TURBO compilation versus SuperBASIC is that byte $54(A6) is negative. Since it turns out that this byte is negative also for SBASIC we have to eliminate that possibility. This is done by looking at -4(A6). This long word is "SBAS" for SBASIC, but not, presumably, for TURBO-compiled programs.

3. Vector variables (#)

In QL World's version of SET/ALTER the code testing an old variable for type (Integer, Float or String), it was assumed that D5.B would contain zero if the variable was a vector (ie had been SET with a preceding "#"). D5 was indeed so set for new variables, (ie for SET) but not for existing ones (ie ALTER). In fact at the code just after LOCK_CASE D5.L is -1 (to indicate ALTER).

In the new version, at LOOKUP_OK, we check whether the code for the variable is READ_PTR. If it is, we are trying to ALTER a vector, otherwise not. We set D5.B zero to indicate a vector and set D5.L negative to indicate an ALTERed vector.

In GOT_NAME we copy D5.B to D4.B changing the latter to 3, to indicate Integer, if it is zero. Later, on at OLD_SCALR, when the value is about to be stored, we check that it is even if D5.L is negative. This prevents vectors being odd, which they must not be.

How To Use SET/ALTER

In case anyone has forgotten how SET/ALTER are used, a brief reminder is given here.

`SET ‹name› TO ‹value›`

The ‹value› should correspond with the type of ‹name›That is ‹value› should be a word-length integer if ‹name› ends with "%", a string if the ending is "$" and a number of any sort otherwise.

If you don't have an assembler, type the following listing into BASIC:

```
100 DEFine PROCedure loader(a$)
110  LOCal x,y,out%,l
120  out%=FOP_OVER(a$)
130  IF out%‹0:PRINT#0, "Can't open "&a$:RETurn
140  RESTORE
150  READ l
160  IF l MOD 2:PRINT#0, "Odd file length":RETurn
170  l=l/2:sum=0
180  FOR x=1 TO l:READ y:WPUT#out%,y:sum=sum+y
190  READ x,y
200  IF sum-2^16*x-y:PRINT#0, "Checksum failure - look at your
      data":CLOSE#out%:DELETE a$:RETurn
210  PRINT#0, "File "&a$&" now ready"
215  CLOSE#out%
220 :
1000 DATA 934
1010 DATA 17402,8,13432,272,20178,2,182,851
1020 DATA 17748,172,1345,19540,17746,0,0,0
1030 DATA 1008,0,0,0,0,29186,24938,8287
1040 DATA 13200,-6144,30723,20085,29190,24924,8287,13840
1050 DATA 28672,20033,9264,12288,8706,26380,12860,2080
1060 DATA 21313,-11134,27386,-7542,13185,-6144,9090,-6142
1070 DATA 30722,20085,29190,24880,8287,13208,-6144,9104
1080 DATA -6142,30722,20085,8287,8272,29187,-11696,2177
1090 DATA 0,24852,-7604,21316,9289,13720,-6144,21642
1100 DATA 20940,-8,30721,20085,10241,18990,84,27146
1110 DATA 3246,21314,16723,-4,26118,12920,282,20113
1120 DATA 8814,88,-27708,11593,88,28672,20085,31487
1130 DATA 24614,10810,-158,26142,20032,29226,10241,10315
1140 DATA 24832,466,9804,17914,-178,9352,17048,12476
1150 DATA 4,636,-9985,31233,18923,16,-17460,26170
1160 DATA 18990,84,27160,3246,21314,16723,-4,26382
1170 DATA 29696,29184,28674,20033,10344,-16,24578,10318
1180 DATA 19077,27418,6707,-6143,26384,2053,7,26370
1190 DATA 31232,4659,-6144,26368,196,28913,20085,19435
1200 DATA 8,11789,12408,278,20112,26352,12337,-6144
1210 DATA 26600,12800,19433,2,29919,-15055,-6142,21129
1220 DATA 21312,26358,5681,-6143,31233,-18884,4,26378
1230 DATA 31235,-18884,5,26370,31234,20032,8300,24
1240 DATA 8236,28,9772,32,13936,-14334,-10301,-19917
1250 DATA -14336,26380,20616,-20344,25326,28921,24576,252
1260 DATA 6195,-14335,-14334,-18379,-6144,26344,15361,21830
1270 DATA 27414,8781,6195,-14334,-14334,21131,21129,-18383
1280 DATA -6144,22222,-16,26316,3120,9,-14336,26374
1290 DATA 28919,24576,198,636,-9985,9799,19435,8
1300 DATA 11784,17914,-412,8304,-14332,10280,2,-18294
1310 DATA 26138,31487,16965,24596,28913,20085,15923,-6142
1320 DATA 28662,18631,-6257,-8530,24,20619,-13764,15
1330 DATA 6149,26114,30723,18564,-10172,-28216,-28476,12392
1340 DATA 280,20112,26324,18996,30720,26484,8308,30724
1350 DATA 3160,20153,26306,-17860,1,26174,18408,4
1360 DATA 8275,8744,-4,23937,-19855,-6144,25634,20032
1370 DATA 14204,16,-8,23681,11787,22920,17402,-556
1380 DATA 13432,218,20114,29185,31487,8814,88,24660
1390 DATA 30721,-10127,-6144,-7604,24602,22664,30722,-17860
1400 DATA 2,26384,30720,19077,27146,2097,0,-6143
1410 DATA 26112,-154,20032,24832,274,636,-9985,20085
1420 DATA 20032,29200,-17860,1,26394,-17860,2,26394
1430 DATA 29196,24598,16890,-644,13432,216,20114,19072
1440 DATA 26126,20085,23681,-11663,-6144,10241,25062,24652
1450 DATA 12410,-674,8708,-19832,25090,29184,-11640,29696
1460 DATA 28696,20033,19072,26290,29712,-28030,17402,-700
1470 DATA 13432,218,20114,8708,24762,28913,24732,2097
1480 DATA 0,-6143,26356,17914,-704,24630,17914,-724
1490 DATA -17860,3,26412,17914,-676,24614,8814,88
1500 DATA 18949,27150,9287,22664,9352,-7604,22404,24576
1510 DATA -154,-17860,1,26054,25298,17914,-694,-26500
1520 DATA 9,-7604,23876,8728,9736,12540,20153,8394
1530 DATA 21253,26126,17896,10,8394,21640,29712,-28030
```

```
1540 DATA 8385,24902,10874,-816,19437,6,-26995,14979
1550 DATA 13366,30722,8302,32,-12094,16880,-6144,29184
1560 DATA 4624,21569,2177,0,17389,2,24578,4824
1570 DATA 20937,-4,16977,8826,-866,-12980,13432,272
1580 DATA 20114,-12980,28672,24576,-270,12529,-6144,21641
1590 DATA 20940,-8,20085
1600 DATA 74, 44245
2000 END DEFine
2010 :
2230 REMark DEFine PROCedure WPUT(ch%,j)
2240 REMark  LOCal m
2250 REMark  m=j DIV 2^8
2260 REMark  IF m< 0:m=256+m
2270 REMark  BPUT#ch%,m:BPUT#ch%,j MOD 2^8
2280 REMark END DEFine
2290 :
```

If ‹name› is preceded by "#" ‹value› must be an even integer. In this case the value returned by the function ‹name› is the contents of the long word at ‹value› offset from the start of the system variables.
Thus

```
SET #job_base TO 104
PRINT HEX$(job_base,32)
```

should result in the hexadecimal value of the start of the job list being printed.
SET operates by making ‹name› a function returning the ‹value› set.

```
ALTER "name" TO ‹value›
```

A ‹name› already set is ALTERed to ‹value› of the same type. The name must be enclosed in inverted commas or apostrophes, but the case of letters is immaterial. If ‹name› had previously been SET using "#" the new value must again be an even integer.

LOADER

The BASIC procedure loader(a$) will produce the assembled binary version of set_asm which you can use if you don't have an assembler, or don't want to type the whole of the assembly program. *[We have included the assembler listing on the cover disk for your convenience - Editor]*

Once the BASIC procedures are entered simply type

```
loader "‹name›"
```

and you have a file ready to be LRESPRd to set up SET and ALTER. For example

```
loader "ram1_set_bin"
LRESPR ram1_set_bin
```

links in the keywords SET and ALTER.

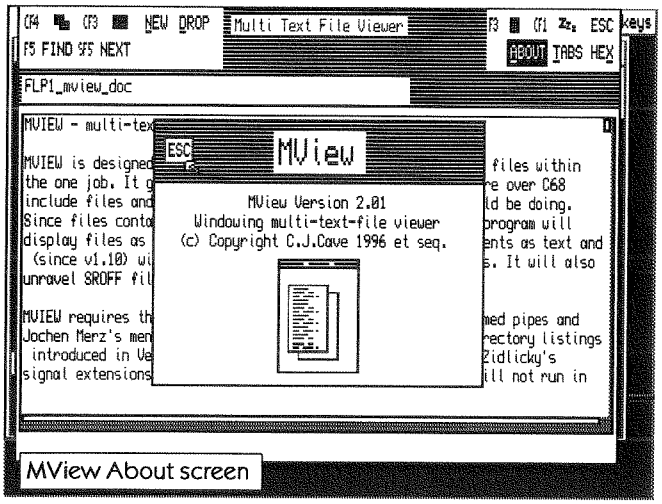
The procedure WPUT has been REMarked out, since most of those wanting to use this version of SET/ALTER presumably have SMSQE, in which case the keyword WPUT will be loaded. If not, simply take out the REMarks and the procedure WPUT will do what's needed. ■

MVIEW - A Multiple File Viewer

A Review

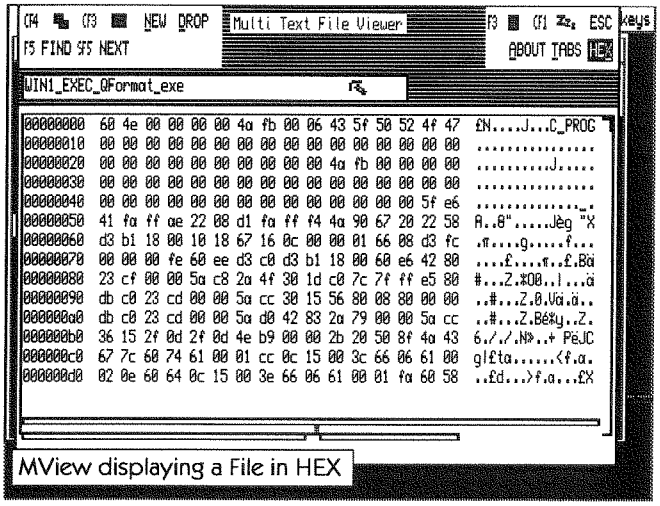
Darren Branagh

About a year ago at the 1998 Byfleet Show, I met Christopher Cave for the first time. Chris had written a file viewer program, and asked Dilwyn to find someone to review it. On Dilwyn's suggestion, Chris gave me a copy of the program which was version 1.19 I believe. As I said, that was a year ago, and I am only reviewing it now - Sorry, Chris.
At the QUANTA AGM in Clevedon several months later, I met Chris again, and he gave me an updated copy of the program, now at version 2.01 - Chris has put considerable effort into this program, and responds to any comments made to enhance it. I think it's a great little program.
The program came about through Chris' need to keep track of various C68 Include files. As every small project, it started as something he needed, and grew into the wonderful program it is today. It now has the ability to load Quill_doc files directly, and will load plain ASCII text files, and even BASIC programs. Text files that are not just pure ASCII can be loaded as HEX instead, and viewed as a Hexdump. Even Qliberator "_sav" files can be converted and viewed as text on SMSQ systems. It will even have a go at displaying SROFF files to a certain degree.
You start MView by EXECing it, it consists of just one file and the manual in Quill_doc format, although Jochen Merz's Menu Extensions are needed, along with ENV_BIN (part of the C68 package) and the Hotkey extensions. You will need to have these pre-loaded to use MView. Two other BASIC programs are included, namely devset_bas (which works around a problem with DEV_USE by setting up the environment variables - I won't go into it here, but it is explained in the manual) and mvdmn_bas which keeping MView running by scanning the job tree for the function NXJOB, if it can't find it - MView is started.



MView About screen

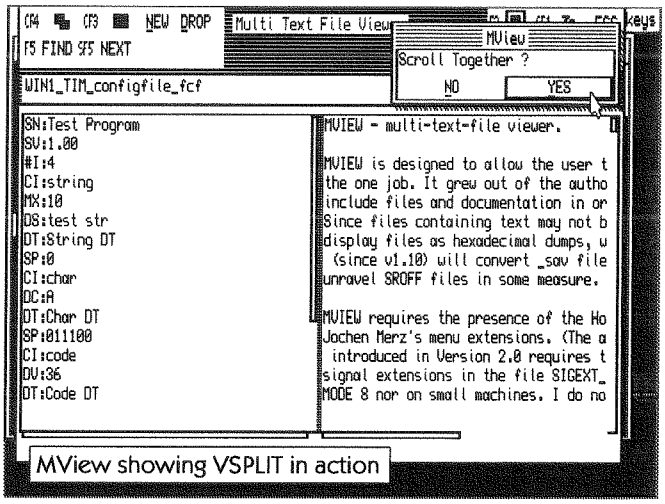
On loading, the default screen layout is presented. It has the commands and clickable icons at the top, with the main text display window taking up most of the screen. Ok, so let's load something - say its own manual, which is in Quill format, and see how it copes. Click on NEW to load a new file, and the familiar FileSelect screen appears. Choose to load the MVIEW_DOC file, and a window appears asking if you want to load it as a Quill document. You can also choose not to, or to load a HEX dump. Choose YES and the manual is displayed as if Quill where loaded, and the formatting of the text is perfect - Ten out of Ten so far.



MView displaying a File in HEX

Chris has wisely kept all the commands in a familiar place - on the F3 key as per quill. You can also access them by clicking on the small green square icon using the mouse. This is where MView comes into a league of its own in my opinion - the sheer amount of ways in which you can view files astounded me. Up to 16 (yes, sixteen) files can be loaded at the same time in any single copy of Mview.

From the Command menu, Choosing the commands HSPLIT or VSPLIT allows you to view files either horizontally or vertically on screen, i.e. either one on top of each other or side by side. You can even choose to scroll both of these files in unison, at the same time, useful for scanning with the eye for similarities or differences between two files.



MView showing VSPLIT in action

The CASCADE command takes it one step further, allowing up to the 16 files to be 'stacked'on top of one another, each in a separate window, so picking each one individually is easy. There is also a command PICK, which displays a list of all the currently loaded filenames, so you can directly choose a file instantly from the list bringing it to the top - this is a very useful feature, and one I use a lot as it saves so much time, provided you can remember all the filenames, of course!

DROP is another nice command, which allows the contents of the stuffer buffer to be loaded, providing it is a filename. MAX allows a file to be maximised to the full size of the screen, if it happens to be windowed, e.g. if you've been using CASCADE. JOIN is useful to add the contents of one file to the other to create on large file. This is useful as you can then scroll through one big file, just by dragging the scroll bar along the edge of the window with the mouse. You will not be able to save this large file, as MView is solely a Viewer, so this large file will be lost when you switch off your QL. Similarly you cannot amend the contents of a file using MView, as this is not what is was designed for - it is worth remembering it is a File Viewer par excellance, nothing more.

However, it does have some text editor type facilities - you can search through a file for a given symbol or string using FIND - and NEXT will present the next occurrence of the

QUANTA

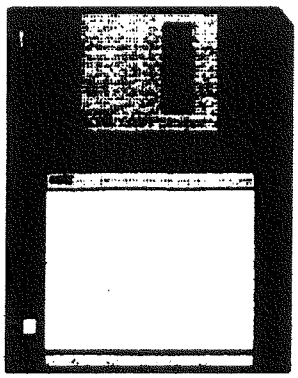
Independent QL Users Group

Worldwide Membership is by subscription only, and offers the following benefits:
Monthly Newsletter - up to 40 pages
Massive Software Library - All Free !
Free Helpline and Workshops
Regional Sub-Groups. One near you?
Advice on Software and Hardware problems
Subscription just £14 for UK members
Overseas subscription £17

Barclaycard: Visa: Access: Mastercard

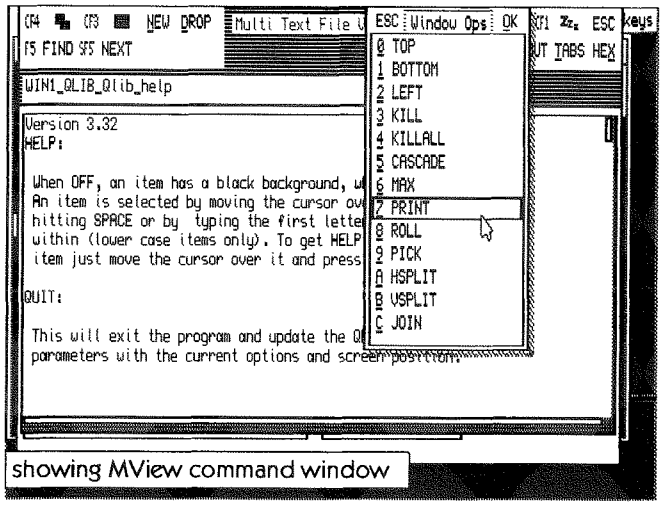
*** Now in our SIXTEENTH successful year ***

Further details from the Membership Secretary



Bill Newell
213 Manor Road
Benfleet
Essex
SS7 4JD
Tel. (01268) 754407

chosen search string. TOP and BOTTOM will bring you to either the start or end of a file, and PRINT will print it.



showing MView command window

The program will work in all the higher screen resolutions I tried - I was running it on a Pentium III under QPC2, and it ran perfectly, ditto on my QXL2 via an Amstrad ALT 286. I also tried it on a Gold Card QL, and although less stunning in 512x256 mode, it worked just fine. Chris mentions in the manual (MVIEW_DOC) that is it prone to hanging in 512x256 mode, although I didn't have this problem. However, the CASCADE option will only work at higher resolutions than 512x256 standard QL mode, and the program will not operate in MODE 8 - not that that will matter to many people. I found the manual to be well written too - the history of the program is clearly defined, and the commands listed in order, with an explanation of each.

All in all, MView is a really useful program if you happen to deal with lots of similar files a lot, or need to keep track of several different updates or different things. The beauty of MView is it can be as simple or as complicated to use as you like - it can be a simple "load two files and spot the difference" type program, right up to one that can handle 16 files at once and display them in several different ways. In its simplest form, it can just be used to read quill docs or ASCII text files or manuals, instead of a text editor or word processor, without the danger of deleting them or resaving them in a different format.

I like MView a lot, and the fact that Chris has made it freeware is a complete boon. He even mentioned to me at Clevedon that he was toying with including the ability to read and display HTML files (Hyper Text Markup Language) to MView in a forthcoming version, so you see work is still continuing on its updating.

I am unsure if it is available from the PD libraries, although it soon should be if not - I will gladly copy it to a disk for anyone at my Q-Celt stand at any QL show. I now find myself having MView on a button on my system by choice - to view a quill_doc file I always use it now, in preference to Quill itself, which might seem odd, but its mainly due to the high level or control I have over the file purely fom the mouse without using the keys, unlike Quill.

Try MView, after a few sessions with it you'll wonder how you ever managed without it - I love it to death. ■

Programmer's Choice: Being Selective - 2

The SuperBASIC SELect structure.

Mark Knight

SELect is a decision making structure in SuperBASIC similar in function to the CASE structure in C, PASCAL and other programming langauges. Many times I have seen IF used in cases where SELect would be clearer, yet SELect is one of SuperBASIC's most useful features. It is easily understood, efficient and flexible, as well as easy to update. There is also a short form of SELect though once again it is best used sparingly. In a large loop where the keyboard is being read you may see something like this:

```
1070    Choice=CODE(INKEY$(#0,-1))
1080    IF Choice=27 THEN RETurn : REMark
        ESCape key
1090    IF Choice>=32 AND Choice<=191 THEN
        Add_CHAR Choice
1100    IF Choice=232 THEN Show_HELP : REMark
        F1
1110    IF Choice=240 THEN Select_COMMANDS :
        REMark F3
```

...this is dreadful SuperBASIC since it involves loads of unnecessary conditions, is slower than a SELect and less readable: so let's now look at SELect. To replace the mess of IFs above we could use:

```
1070    Choice=CODE(INKEY$(#0,-1))
1080    SELect ON Choice
1090      =27
1100        REMark ESCape key
1110        RETurn
1120      =32 TO 191
```

```
1130        Add_CHAR Choice
1140      =232
1150        REMark F1
1160        Show_HELP
1170      =240
1180        REMark F3
1190        Select_COMMANDS
1200    END SELect
```

This is a little longer but much clearer, and it will work faster than the mess of IF lines in the previous example. SELect works by choosing a series of SuperBASIC lines depending upon the value of a variable, the one specified in the opening SELect line. Later lines can set lists of values or ranges of values defined in a very similar way to those used in a FOR loop; more about this later.

Not only does SELect as a whole have a long and short form, but within the long form there are long and short form SELect lines. This is often a source of confusion, so I'll try and explain it a little here. In the definition of SELect the programmer chooses a variable to be tested, in the above example "Choice". Between the SELect and the END SELect there are then lines to divide up the program, separating the program statements according to the value of the variable. These may take several forms, like this:

```
1070    Choice=CODE(INKEY$(#0,-1))
1080    SELect ON Choice
1090      ON Choice=27
1100        REMark ESCape key pressed.
1110        RETurn
1120      ON Choice=32 TO 191
1130        Add_CHAR Choice
1140    END SELect

1070    Choice=CODE(INKEY$(#0,-1))
1080    SELect ON Choice
1090      =27
1100        REMark ESCape key pressed.
1110        RETurn
1120      =32 TO 191
1130        Add_CHAR Choice
1140    END SELect
```

...or even short form lines like these:

```
1070    Choice=CODE(INKEY$(#0,-1))
1080    SELect ON Choice
1090      ON Choice=27 : RETurn : REMark ESCape
          key pressed.
1100      ON Choice=32 TO 191 : Add_CHAR Choice
1110    END SELect

1070    Choice=CODE(INKEY$(#0,-1))
1080    SELect ON Choice
1090      =27 : RETurn : REMark ESCape key
          pressed.
1100      =32 TO 191 : Add_CHAR Choice
1110    END SELect
```

Notice that the lines using the "ON Choice=" will take longer to type but will not add anything to the function of the program. In fact these lines aren't even properly checked, so we could input a program like this:

```
100 REPeat Program
110   TestVal=RND(-100 TO 100)
120   SELect ON TestVal
130     ON Rubbish=-100 TO -50
140       PRINT"Very low!"
150     ON SomeOtherRubbish=-49 TO -1
160       PRINT"Low"
170     ON DoesntExist=0 TO 49
180       PRINT"Moderate"
190     ON NaNaNa=50 TO 100
200       PRINT"High"
210   END SELect
220 END REPeat Program
```

...and the program will run without presenting an error message, skipping over the silly variable names used in the listing! Since it isn't checked properly why type in the SELect variable repeatedly and risk typing errors messing up your program? SBASIC is more sensible and will produce an error message in the above case, but why bother with the extra typing?

SELect allows far more than the two choices offered by IF and is therefore more powerful and potentially more confusing. Ranges in SELect can be single numbers, ranges of numbers or just like FOR loops, multiple lists of numbers and ranges separated by commas. To see this in action try the following (I confess this is not a useful example, but it works):

```
100 CLS : CLS#2
110 REPeat Program
120   Example=RND(0 TO 255)
130   SELect ON Example
140       =1,3,5
150         PRINT"Wibbla!"
160       =2,4,6 TO 22
170         PRINT"Stop it that hurts!"
180       =23 TO 27,30 TO 35
190         PRINT"IG!"
200       =36 TO 99
210         PRINT"Ouch!"
220   END SELect
230   PRINT Example
240 END REPeat Program
```

I have deliberately left out much of the range, so line 120 gives the variable "Example" a range of 0 to 255, while 0 and the values from 100 to 255 are missing from the SELect structure's list of choices: this means all that happens when the variable is set to these values in line 120 is that they will be PRINTed at line 230. Any values not covered in a SELect range will result in the statements after the END SELect being executed.

To cover cases where we always want something done whatever the value of the SELect variable, SELect has a superb additional feature; the REMAINDER option. To use it, change the above example to:

```
100 CLS : CLS#2
110 REPeat Program
120   Example=RND(0 TO 255)
130   SELect ON Example
140      =1,3,5
150         PRINT"Wibbla!"
160      =2,4,6 TO 22
170         PRINT"Stop it that hurts!"
180      =23 TO 27,30 TO 35
190         PRINT"IG!"
200      =36 TO 99
210         PRINT"Ouch!"
220      =REMAINDER
230         PRINT"Yah, missed!"
240   END SELect
250   PRINT Example
260 END REPeat Program
```

What the REMAINDER directive does is to check for any values not covered in the rest of the SELect structure and take the action specified by the programmer. REMAINDER is a very powerful "catch-all" for the SELect clause, not always needed but a great boon to programmers.
What hasn't been made clear so far is that the standard SuperBASIC SELect uses floating point values, so the following works:

```
100 CLS
110 FOR Test=1 TO 10
120   TestVal=RND
130   SELect ON TestVal
140     =0 TO .2499999
150       PRINT "Less than a quarter.";
160     =.25 TO .4999999
170       PRINT"At least a quarter, less than
          a half.";
180     =.5 TO .7499999
190       PRINT"At least a half, less than
          three quarters.";
200     =.75 TO 1
210       PRINT"More than three quarters, up
          to one.";
220     =REMAINDER
230       PRINT "Something has gone wrong.";
240       STOP
250   END SELect
260   PRINT!TestVal
270 END FOR Test
```

As an aside, notice the use of a REMAINDER clause which in this case should never be called; this is often a useful device to catch typing or programming errors during the development of a program. The program prints ten lines of information about the random values it chooses, each using the floating point SELect range specified.

There are several things to note about SELect and the values and ranges it uses.
The first thing to remember is that single values are evaluated approximately in case calculations that lead to them contain rounding errors. The approximation used is itself approximate, it is around one part in ten million. To see this in action try the following listing:

```
100 CLS
110 FOR Example=2E9-500 TO 2E9 STEP 10
120   SELect ON Example
130     =2E9
140       PRINT Long_INTEGER$(Example);"
          Matched with ";Long_INTEGER$(2E9)
150     =REMAINDER
160       PRINT Long_INTEGER$(Example);" Yah,
          missed!"
170   END SELect
180 END FOR Example
190 :
200 DEFine FuNction Long_INTEGER$(Tk_Number)
210   LOCal Tk_Negative%,Tk_BigBit,
      Tk_LittleBit
220   IF Tk_Number<1000 AND Tk_Number>-1000
      THEN RETurn Tk_Number
230   Tk_Negative%=0
240   IF Tk_Number<0 THEN
250     Tk_Negative%=1
260     Tk_Number=ABS(Tk_Number)
270   END IF
280   Tk_BigBit=INT(Tk_Number/1000)
290   Tk_LittleBit=Tk_Number-(Tk_BigBit*1000)
300   IF Tk_Negative%=0 THEN
310     RETurn  Long_NUMBER$(Tk_BigBit)&
        Leading_ZERO$ (Tk_LittleBit,3,"0")
320   ELSE
330     RETurn "-"&Long_NUMBER$(Tk_BigBit)
        &Leading_ZERO$(Tk_LittleBit,3,"0")
340   END IF
350 END DEFine Long_INTEGER$
360 :
370 DEFine FuNction Long_NUMBER$(Tk_AnyNumber)
380   LOCal Tk_No$(28),Tk_An$(28), Tk_Bn$(28),
      Tk_Ex$(28),Tk_Dec%
390   LOCal Tk_Exx%,Tk_Neg%,Tk_Start$(3)
400   Tk_No$=Tk_AnyNumber
410   IF "E" INSTR Tk_No$=0 THEN
420     IF "." INSTR Tk_No$=1 THEN RETurn
        "0"&Tk_No$
430     IF "-." INSTR Tk_No$=1 THEN RETurn
        "-0"&Tk_No$(2 TO LEN(Tk_No$))
440     RETurn Tk_No$
450   END IF
460   Tk_Dec%="." INSTR Tk_No$
470   Tk_Exx%="E" INSTR Tk_No$
480   IF Tk_Dec%>0 THEN
490     Tk_An$=Tk_No$(1 TO Tk_Dec%-1)
500     Tk_Bn$=Tk_No$(Tk_Dec%+1TO Tk_Exx%-1)
510   ELSE
520     Tk_An$=Tk_No$(1 TO Tk_Exx%-1)
530     Tk_Bn$=""
540   END IF
550   Tk_Ex$=Tk_No$(Tk_Exx%+1 TO LEN(Tk_No$))
560   IF LEN(Tk_Bn$)<=Tk_Ex$ THEN
570     RETurn Tk_An$&Tk_Bn$&FILL$
        ("0",Tk_Ex$-LEN(Tk_Bn$))
```

```
580   END IF
590   Tk_Neg%="-" INSTR Tk_Ex$
600   IF Tk_Neg%>0 THEN
610     Tk_Ex$=Tk_Ex$(2 TO LEN(Tk_Ex$))
620     Tk_Neg%="-" INSTR Tk_An$
630     IF Tk_Neg%>0 THEN
640       Tk_An$=Tk_An$(2 TO LEN(Tk_An$))
650       Tk_Start$="-0."
660     ELSE
670       Tk_Start$="0."
680     END IF
690     RETurn Tk_Start$&FILL$("0",Tk_Ex$-1)
      &Tk_An$&Tk_Bn$
700   END IF
710 END DEFine Long_NUMBER$
720 :
730 DEFine FuNction Leading_ZERO$ (Tk_AnyNum$,
    Tk_Figures%,Tk_Lead$)
740   IF LEN(Tk_AnyNum$)<Tk_Figures% THEN
750     RETurn FILL$(Tk_Lead$,Tk_Figures%-LEN
        (Tk_AnyNum$))&Tk_AnyNum$
760   ELSE
770     RETurn Tk_AnyNum$
780   END IF
790 END DEFine Leading_ZERO$
800 :
```

...because the FOR loop variable is such a large number it is matched only approximately, and not just one but several of the values match with the value choosen in line 130. It is a side issue here but interesting to note that SuperBASIC can handle ten digit numbers as long as they are within a certain range. Long integers in the range -2,147,483,648 to 2,147,483,647 can be used though they can't be PRINTed without some trickery: The SuperBASIC FuNctions in the listing provide the required trickery.

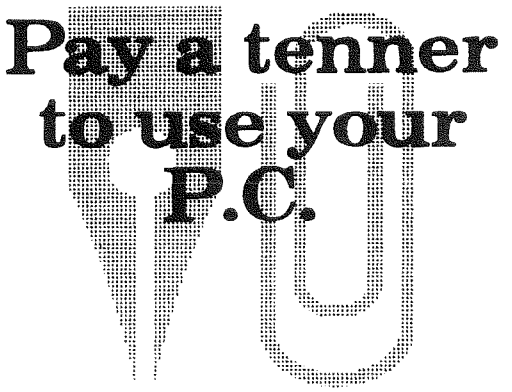
The second thing to note is that ranges are matched exactly and are inclusive, so in a case where a SELect contains the following line:

```
10100   =64 TO 100
```

...any value from 64 to 100 will match, including such values as 64.01 or 75.5 or whatever: this means that approximate calculations may miss the SELect range.

It is also important to remember that there is no equivalent to the FOR loop STEP so ranges include all values, and must run from lowest value to highest. On a Sinclair ROM QL the following example program will print "Missed" followed by the value of the variable "Example" ten times, even though every time the value is within the range specified in line 140:

```
100 CLS
110 FOR Test=1 TO 10
120   Example=RND(1 TO 10)
130   SELect ON Example
140     =10 TO 1
150       PRINT"OK"!Example
160     =REMAINDER
```

At JUST WORDS! we are not paranoid about the PC. It is there and many of us have to use it daily. We have therefore written the most versatile possible utility to transfer files between your QL and PC word processors. Your QL word processor can be Quill, Perfection or Text87, and your PC word processor either Windows or Dos based. The features of this pointer driven program include:

Correct transfer of accented and similar characters (of special interest to non-english language QL users).

Possible transfer of bold, underlined and italic text (depending on your word processing software).

Setting font, font size, justification, margins and tabs on the QL for windows based word processors.

Experimental e-mail reading and OCR-text tidy routines.

QL-2-PC TRANSFER costs just £10 or 15 Euros. Its is available either directly from JUST WORDS! or from QBRANCH.

Payment can be made by sterling cheque or giro transfer in Euros to Netherlands Postbank number 4111942 (G.T. Wicks).

Geoff Wicks, 28 Ravensdale, Basildon, Essex SS16 5HU, United Kingdom.

Telephone: +44 (0)1268 - 281826

e-mail: geoffwicks@hotmail.com

```
170       PRINT"Missed"!Example
180    END SELect
190 END FOR Test
```

Unless you change line 140 so that the range is "1 TO 10" it will give an error in SBASIC, as it would in SuperBASIC if things were ideal.

Short form SELect:

This is a simple single line structure that is ideal for testing a single
range, for example to replace the following IF:

```
100 CLS
110 FOR Test=1 TO 10
120   Example=RND(0 TO 11)
130   IF Example›=3 AND Example‹=7 THEN PRINT
      "In range"
140 END FOR Test
```

...we could use:

```
100 CLS
110 FOR Test=1 TO 10
120   Example=RND(0 TO 11)
130   SELect ON Example=3 TO 7 : PRINT "In
      range"
140 END FOR Test
```

This is simpler to type and easier to read than the equivalent IF, though it does not allow for the refinement of an ELSE clause.

Minerva and SBASIC integer SELect:

Both Minerva and SBASIC allow integer SELect, so the following will work:

```
100 CLS
110 FOR Test=1 TO 20
120   Example%=RND(0 TO 11)
130   SELect ON Example%
140     =1 TO 10
150       PRINT"OK"!Example%
160     =REMAINDER
170       PRINT"Missed"!Example%
180   END SELect
190 END FOR Test
```

Integer SELect is faster, so where keyboard codes or other valid integers are involved it pays to use the facility.

Turbo and Q-Liberator:

Turbo and Q-Liberator will allow integer SELect even on Sinclair ROM QLs by using compiler directives. For Q-Liberator the following will make the "Example" variable into an integer:

```
100 CLS
110 DEF_INTEGER Example
120 FOR Test=1 TO 20
130   Example=RND(0 TO 11)
140   SELect ON Example
150     =1 TO 10
160       PRINT"OK"!Example
170     =REMAINDER
180       PRINT"Missed"!Example
190   END SELect
200 END FOR Test
```

Turbo works in a similar way, for Turbo change line 110 to:

```
110 IMPLICIT% Example
```

Turbo also has an extra trick: string SELect as well! For this we would use the following:

```
100 CLS
110 IMPLICIT$ Example
120 FOR Test=1 TO 20
130   Example=CHR$(RND(32 TO 191))
140   SELect ON Example
150     =" "
160       PRINT"A space!"
170     =REMAINDER
180       PRINT"A visible character; ";Example
190   END SELect
200 END FOR Test
```

A disadvantage is that this example can't be tested under the interpreter and must be compiled before it will work. Fragments from a program using string SELect might be:

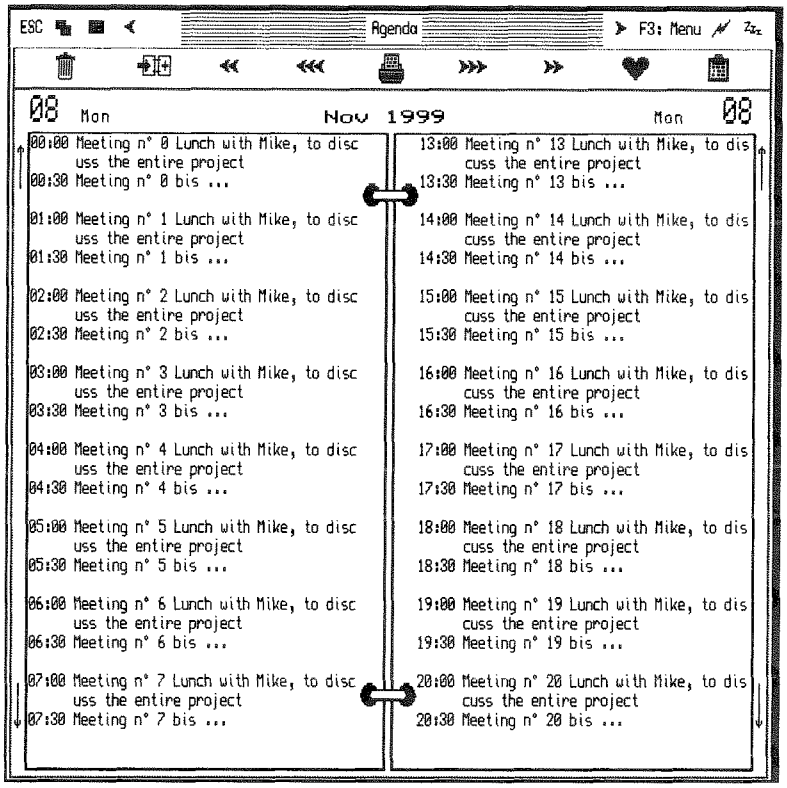
```
150 IMPLICIT$ TestWord

10320     SELect on TestWord
10330       ="Out"
10340         Do_OUT TestParam$
10350         RETurn
10360       ="In"
10370         Do_IN TestParam$
10380         RETurn
10390       =REMAINDER
10400         Syntax_ERROR
10410     END SELect
```

Turbo's string SELect also uses numeric ranges not string ones; for more details see the later version of the Turbo manual which gives information in section 4.1.4.6 (Page 81 in the printed version). Minerva also allows string SELect though it works slightly differently in some circumstances to the Turbo string SELect. This is only important if you are writing a program on a Minerva system, as you have to remember that you can't test your program under the interpreter. It may appear to work well, but the different behaviour when compiled requires seperate testing. If you aren't using Minerva string SELect won't work anyway until the program is compiled so you can't make the same error.

Agenda

Wolfgang Lenerz

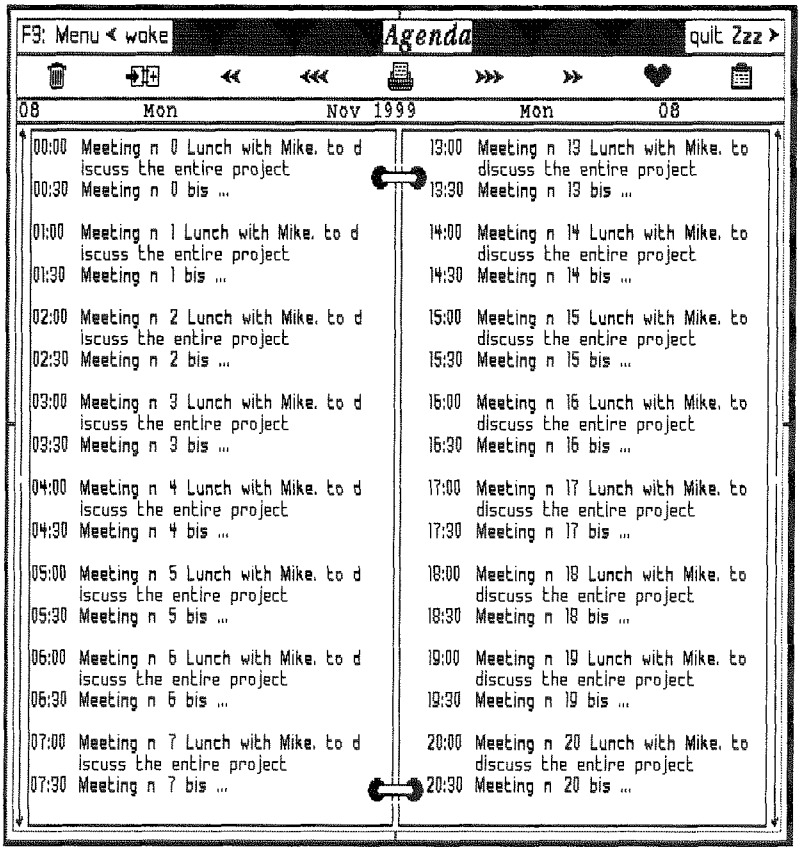
New out from Jochen Merz Software is Agenda. As the name implies, this is an appointments calendar, it helps you to keep track of your appointments. I'll come clean right away and tell you that I'm the author of Agenda, so, of course, you'll take all laudatory comments on the software with a pinch (nay, a baleful) of salt (though I'll try to keep this as objective as possible).



As with all of my software that has gone commercial (such as FiFi and the Basic Linker), Agenda originally came about because I needed a program quite like it. I had something very much simpler and cruder in use for quite a number of years as part of my professional software suite. Over the time, I added more and more features until one day I decided to re-write it entirely and the fully-fledged Agenda, as now on sale, is the result of this.

As mentioned above, Agenda is an appointments calendar. It is modelled on a day planner (see figure 1, the main window - please note that I have put some rather silly test data into that...), which means that it will show you your appointment for one day. Agenda will keep your appointments for you and remind you of them. You can add new appointments, delete or modify them and see them in all details. Agenda uses the well-proven DATAdesign database engine for this, but, of course, this is completely transparent to the user.

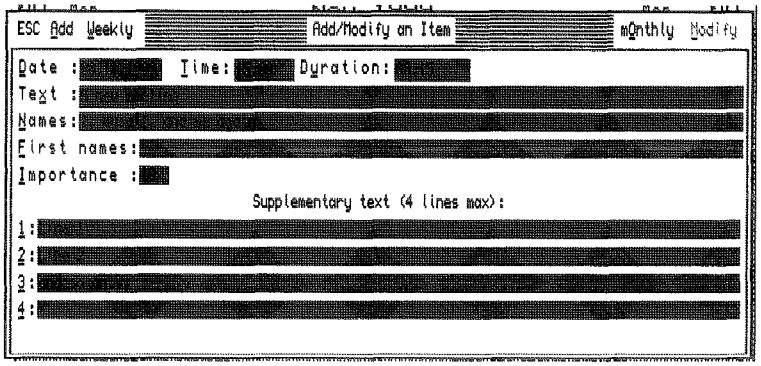
Agenda actually comes in two flavours: One runs under the "normal" Pointer Environment (which is what you see in figure 1), the other runs under Prowess (see figure 2, the same window in the Prowess version). During installation of the software, you can choose which one you want. For each of these two versions, you can choose english or german language versions (I could make french and spanish versions if there was a demand for it). To make things easier, I've provided a very simple boot that copies the correct files for the version you need from the floppy to wherever you want it.



The main Agenda screen (see fig. 1) shows you one day, and the appointments you have for

that day. Using the approprate icons, or through the F3 Menu, you can easily go from this day to the next or previous day, week or month. Or you can go to any date you like. All of these actions (and more) are achieved by clicking on the icons located, mostly, in the lower menu bar. For the mousely challenged, or for those who simply don't like to use rodents or having their screen full of icons, you can remove the menu bar. This can be done either at runtime, or during configuration, but it is always possible to show/hide the menu bar again at runtime. This might prove especially useful for those who don't use emulators nor newer machines, and still have the original QL screen.

The data is displayed in two sections, one for the morning (on the left) and the other for the afternoon (on the right). You can also set importance levels (from 1 to 10), and only display those appoinments or meetings that are really important to you that day.



There is a provision for recurring appointments: you can determine that a certain appointment is repeated every week or month for as long as you like. This is useful for recurring things, such as payment of your rent. Agenda also provides a mechanism for yearly recurring events, such as birthdays: there is a "birthdays and specials" list. It can show you whether any of these events is today. So, you have no excuse for forgetting your mother-in-law's birthday any more.

Agenda also keeps a Todo list, on which you can put things you have to do. These are carried over till tomorrow until you definitely strike them from the list. You not only have you no excuse not to remember you mother-in-law's birthday, you also no longer have any excuse for not sending her flowers!

Entering/modifying meetings is done via a detailed entry form (see figure 3). Here, you can set the dates and times for your meetings, as well as the text for it. You can also set its importance level and, of course, the name of the people you might be meeting. Finally, there are four more lines of text (each can be as long as 1000 characters), which you can use to store additional information, or perhaps comments on the appointmant once it has taken place.

You are warned when a new meeting will overlap with one that already exists. Agenda doesn't stop you from entering meetings that overlap since you are free to do what you like, but at least it warn you that a potential time conflict exists.

Finally, you can also remove from the database all older data in one go. This purge is quite important if you want to avoid the database getting too large over the years.

All I all, I believe (and I hope) that this is an emminently useable piece of software. As already mentioned, I use it day in, day out. I also know from other users that they find it useful.

Gee Graphics! (on the QL?) - part 13

Herb Schaaf

Three more ECT's

This time we add in three more 2 Dimensional "ECT's". They are aspECT, projECTion, and perspECTive.

Merge the listing

`QL_SheaReflectAdd3Ects_bas`

in GG#13 with the previous listing

`QL_SheaReflectLess3_bas`

from GG#12 to add transformations in 2D that will let you change the aspect ratio, and illustrate parallel and perspective projection.

The 1st transformation is selective scaling along a single axis that will deform the graphic object by changing the aspect ratio. You might think of it as stretching or shrinking the object along the selected axis.

The other 2 graphic transformations are parallel projection and perspective projection. In 2D this simply finds a point on the reference line that is mapped to it from the object point. For parallel projection this point is RETurned from the FuNction project_on_line (x,y,A,B,C) as (x_pt, y_pt).

For perspective projection we establish a reference point (viewpoint) as well as a reference line. There are three situations depending on the relationship of the object point, the

RWAP SOFTWARE

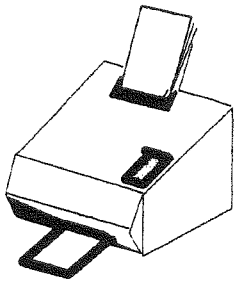
FASTER, FINER, BRIGHTER

EPSON (ESC/P2) PRINTER

DRIVERS FOR ProWesS

180 to 720 DPI

(FREE UPGRADE TO 1440 DPI)



£8

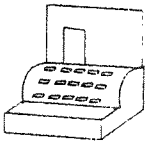
Q-Route

v1.08B

£25

Take the strain
Out of Travel

RWAP Software, 4 Anvil Crescent,
Coseley, West Midlands
WV14 8GA
TEL: 01902 836888
E-MAIL: rwap@qbranch.demon.co.uk



Cheques in £sterling
payable to 'R.Mellor'

Q-Route Maps

Britain v1.08 (included)
BIG Britain v2.01 £5
Ireland v1.09 £5
Yorkshire v1.02 £1
Belgium v1.01 £2
Catalonia v1.01 £2

image point (on the reference line), and the reference point. We assume they are colinear, and not all at the same place. Based on which of the 3 points is in-between (in the middle) we have the following concepts:

1 - the reference point is in the middle - A 3D example is the pinhole camera, where the pinhole is the reference point, the reference line is on the film plane, and the object point is on the photographer's subject. Light from a point on the object passes through the pinhole (reference point) and strikes the film at the image point. Another 3D example is the human eye. Our pupil is the reference point, our retina is the image point, and whatever we see is our collection of object points. Actually, as you know, it is even more complicated and wonderful because of our two eyes, variable lenses, brains, experiences, imaginations, etc.

2 - the image point is in the middle - Here is another 3D example - Imagine looking out through a window screen at a distant object. The line from our eye (the reference point) to a selected point on the distant object passes through the window screen at the image point.

3 - the object point is in the middle - We might think of the reference point as a point source of light that illuminates the object point and casts a shadow on the reference line. Think of a candle (reference point) casting a shadow of your hand (object points) onto the wall (image points) as a 3D example.

The menus and such are as before, with different colors used to indicate negative or positive distance, and which of the 3 is the in-between item.

Next time we hope to begin extending these 2D transformations into the 3D world, from the line to the plane. It has helped me to do them in 2D first, and I hope it will make it easier to carry the concepts over into the higher dimension.

```
100 REMark QL_SheaReflectAdd3Ects_bas for GG#13 H.L. Schaaf September 10, 1999
240  IF transform$=='p' :  INK  0 : project_points_to_line
250  IF transform$=='v' :  INK  0 : transform
260  IF choice_made < 7 THEN
1500    = 6 : array(i,axis_num) = array(i,axis_num) * changeaspect_factor
1510    = 7 : IF (i) THEN
1520           x_pt = project_on_line(array(i,1),array(i,2),A,B,C)
1530           p_r =  Point_to_line2D(array(i,1),array(i,2),A,B,C)
1540           INK 2*SGN(p_r) + 4
1550           FILL 1 : CIRCLE x_pt, y_pt,vert_scale/128 : FILL 0
1560          END IF
1570    = 8 : Av = view_line (x_ch, y_ch, array(i,1), array(i,2))
1580          x_i = intersect_2lines (A,B,C,Av,Bv,Cv)
1590          IF NOT (lines_parallel) THEN
1600 REMark reference point to reference line distance
1610           p_r =  Point_to_line2D(array(i,1),array(i,2),A,B,C)
1620 REMark compare with reference point to reference line distance
1630 REMark what we want to know is which is the middle point
1640 REMark if object point ; shadow cast on wall by a light source
1650 REMark if image point ; view through a window screen
1660 REMark if reference point ; reversal like a pin hole camera
1670 REMark get three distances using dist_btwn
1680 REMark find maximum distance; it tells which item is "in-between"
1690           DIM rp_im_ob(3)
1700           rp_im_ob(1) = dist_btwn(x_i,y_i,array(i,1),array(i,2))
1710           rp_im_ob(2) = dist_btwn(x_ch,y_ch,array(i,1),array(i,2))
1720           rp_im_ob(3) = dist_btwn(x_i,y_i,x_ch,y_ch)
1730           max_dist = 0
1740           FOR j = 1 TO 3
1750            IF rp_im_ob(j)>max_dist THEN
1760             max_dist = rp_im_ob(j)
1770             in_between = j
1780            END IF
1790           END FOR j
1800           SELect ON in_between
1810            = 1 : INK 242   : REMark pinhole camera and film
1820            = 2 : INK 2     : REMark image at window screen
1830            = 3 : INK 0     : REMark shadow cast on wall by light
1840            = REMAINDER : PRINT#0;"Error with rp_im_ob comparison"
1850           END SELect
1860           FILL 1:  CIRCLE x_i, y_i,vert_scale/128  :FILL 0
1870          END IF
1920 :
```

```
2440   PRINT#0,"[A]spect ratio",,"[P]rojection to line",,"[V]iewpoint"
2540   IF transform$ == 'a' : choice_made = 6
2550   IF transform$ == 'p' : choice_made = 7
2560   IF transform$ == 'v' : choice_made = 8
2630   = 4, 5, 7 : Set_reference_line
2640   = 6 : changeaspect_menu
2650   = 8 : perspective_menu
3990 :
4000 DEFine PROCedure changeaspect_menu
4010  CLS#0 : PRINT #0;"Change scale along [X] or [Y] axis ?"
4020  axis$ = INKEY$(-1): axis_num = (CODE(axis$)-87) MOD 32
4030  IF axis_num › 2 OR axis_num ‹ 1 : GO TO 4010
4040  CLS#0 : PRINT#0;"a factor greater than 1 will stretch the object";
4050  PRINT#0;" along the ";axis$;" axis"
4060  PRINT#0;"a factor less than 1 will shrink the object";
4070  PRINT#0;" along the ";axis$;" axis"
4080  INPUT #0;"ENTER factor to change aspect : ",changeaspect_factor
4090  transform
4100  IF changeaspect_factor == 1 : change$ = " unchanged by "
4110  IF changeaspect_factor  ‹ 1 : change$ = " shrunk by "
4120  IF changeaspect_factor  › 1 : change$ = " stretched by "
4130  CLS#0:PRINT#0\\,,axis$;" axis";change$;changeaspect_factor
4140 END DEFine changeaspect_menu
4150 :
4160 DEFine PROCedure project_points_to_line
4170  i_nk = 7 : show_reference_line : i_nk = 0 : transform
4180  CLS#0 : PRINT #0\\,,"object points projected to reference line"
4190  PRINT #0\,,"  touch [spacebar] to continue" : PAUSE :CLS#0
4200 END DEFine project_points_to_line
4210 :
4220 DEFine PROCedure perspective_menu
4230  CLS#0
4240  set_cross_hairs
4250  Set_reference_line
4260  vp_to_rl = Point_to_line2D (x_ch,y_ch,A,B,C)
4270  i_nk = 0 : show_reference_line
4280  INK 7 :transform
4290  CLS#0 : PRINT #0\\,"  line through viewpoint and object projected
 to reference line"
4300  PRINT #0\,,"  touch [spacebar] to continue" : PAUSE :CLS#0
4310 END DEFine perspective_menu
4320 :
4330 DEFine PROCedure set_cross_hairs
4340  x_ch = 0 : y_ch = 0 : ch = 16 : INK  7 :  OVER  -1
4350  REPeat set_crosshairs
4360   CLS#0 : PRINT#0;"Use arrow keys to position viewpoint"
4370   PRINT #0;"touch + or - keys to change size of crosshairs"
4380   PRINT#0;"use with SHIFT for finer adjustments, ";
4390   PRINT#0;" when satisfactory, touch ENTER"
4400   PRINT #0\"Now at X = ";x_ch;", Y  = ";y_ch;", size = ";ch;"      "
4410   LINE x_ch,y_ch-ch TO x_ch,y_ch+ch :LINE x_ch-ch,y_ch TO x_ch+ch,y_ch
4420   change_cross_hair = CODE(INKEY$(-1))
4430   LINE x_ch,y_ch-ch TO x_ch,y_ch+ch :LINE x_ch-ch,y_ch TO x_ch+ch,y_ch
4440   SELect ON change_cross_hair
4450    = 61  : ch = ch + 10
4460    = 43  : ch = ch +  1
4470    = 45  : ch = ch - 10
4480    = 95  : ch = ch -  1
4490    = 208 : y_ch = y_ch + 10
4500    = 212 : y_ch = y_ch +  1
4510    = 216 : y_ch = y_ch - 10
4520    = 220 : y_ch = y_ch -  1
4530    = 200 : x_ch = x_ch + 10
4540    = 204 : x_ch = x_ch +  1
4550    = 192 : x_ch = x_ch - 10
4560    = 196 : x_ch = x_ch -  1
4570    =  10 : EXIT set_crosshairs
4580    = REMAINDER  : GO TO 4360
4590   END SELect
4600  END REPeat set_crosshairs
```

```
4610  OVER 0
4620  LINE x_ch,y_ch-ch TO x_ch,y_ch+ch :LINE x_ch-ch,y_ch TO x_ch+ch,y_ch
4630 END DEFine set_cross_hairs
4640 :
4650 DEFine FuNction view_line(x1,y1,x2,y2)
4660  view_is_vertical = 0 : view_is_horizontal = 0
4670  IF x1 = x2 : Av = 1 : Bv = 0  : Cv = -x1 : view_is_vertical = 1
4680  IF y1 = y2 : Av = 0 : Bv = 1  : Cv = -y1 : view_is_horizontal = 1
4690  IF NOT((view_is_vertical)OR(view_is_horizontal)) THEN
4700   Av = 1 : Bv =  -(x2-x1)/(y2-y1)
4710   cv1 = Av*x1 + Bv * y1  : cv2 = Av*x2 + Bv * y2
4720   IF (cv1==cv2)OR(NOT(cv1 AND cv2)) THEN
4730    Cv = -cv1
4740   ELSE
4750    PRINT #0;cv1,cv2,"check your work !" :PAUSE
4760   END IF
4770  END IF
4780 RETurn Av : RETurn Bv : RETurn Cv
4790 END DEFine view_line
4800 :
4810 DEFine FuNction intersect_2lines(A1,B1,C1,a2,b2,c2)
4820  lines_parallel = 0  : x_i = 0 : y_i = 0
4830  IF (NOT(B1) AND NOT(b2)) THEN
4840   lines_parallel = 1
4850  ELSE
4860   IF ((B1)AND(b2)) THEN
4870    IF ((A1/B1)==(a2/b2)): lines_parallel = 1
4880    IF ((lines_parallel) AND ((C1/B1)==(c2/b2))): lines_parallel = 2
4890   END IF
4900  END IF
4910  IF NOT(lines_parallel) THEN
4920   x_i = -(b2*C1-B1*c2)/(b2*A1-B1*a2)
4930   y_i = -(a2*C1-A1*c2)/(a2*B1-A1*b2)
4940  END IF
4950  RETurn x_i : RETurn y_i
4960 END DEFine intersect_2lines
4970 :
4980 DEFine FuNction dist_btwn(xpt,ypt,x,y)
4990 REMark distance between two points xpt,ypt as point of origin
5000  xdis = (x-xpt) : ydis = (y-ypt)
5010  IF ((ABS(xdis)›1E308)OR(ABS(ydis)›1E308)):PRINT #0;'overflow possible!'
5020 REMark if ABS xdis or ydis › 1E308, will have overflow error message
5030  sqdist = ((xdis*xdis)+(ydis*ydis))
5040  dbtw = 0
5050  IF (sqdist) : dbtw = SQRT(sqdist)
5060  RETurn dbtw
5070  RETurn xdis
5080  RETurn ydis
5090 END DEFine :REMark FN dist_btwn(xpt,ypt,x,y)
5100 :
5110 REMark end of listing QL_SheaReflectAdd3Ects_bas
```

Two Configuration Solutions

Al Boehm

Geoff Wicks in "You and Your Software - Just good Friends? Part 4 - Problems and solutions." in the Sept./October 1999 QL Today writes about the problem of configuring a program. I ran into this problem head-on when I wrote MIDIPlayer - a QL program that plays Standard MIDI files using Simon N. Goodwin's DIY_MIDI keywords. I set as a goal that it would run on all the QL Operating Systems including QDOS, Minerva, and SMSQ/E. Also that it would run compiled and in interpreted SuperBasic for those who wanted to add their own bell and whistles (In MIDI you actually can add bells and whistles!)

I wanted a user to be able to configure the program to look good on their system. I wanted to be able to change such things as paper and ink color and size of type for those with bad eyes like me or if it is run on a TV. Other MIDI programs that are planned will require a screen size option, but MIDI-Player doesn't use the screen that much so I opted for simplicity and kept it at the standard 512 by 256.

Well how does one pass a set of configuration parameters to either a Qliberated or Turbo compiled program and to SBasic/SuperBasic and in all QL operating systems?
Passing a string of configure parameters after the Exec filename looks arcane and would require a lengthy explanation in the User Manual. Plus, I not sure how to do this so it would work on all OS and with SuperBasic. (But see below.)
Using a config block has the drawback that you need to write a config program! One more chore, plus the task of explanation in the manual.

SET

I came across the SET keyword in vol. U of Simon's DIY keywords available from QUBBESOFT P/D. These are not quite freeware but all that is required is that you send a postcard to Simon. SET allows a name to be given a value. From then on, the name acts like a function but doesn't require parenthesis. For example,

```
120 SET paper_color TO 2 . . .
5250 PAPER#3, paper_color
```

will cause the paper color for window #3 to be 2. For configuring, the SET can be in the boot program while it's use can be in a SuperBasic or in a Qliberated or Turbo compiled program. If the name is chosen to be descriptive, hardly any explanation is required. Plus while not everyone is a programmer, I think QL users know how to change a number in a boot program.
SET can also be used with strings. So one possible solution to the language problem would be to use SET for error and prompt messages:

```
150 SET Enter_Age$ TO "Enter age of client"
160 REMark No more then 22 characters in Enter_Age$ to fit format. ...
1220 INPUT (Enter_Age$), age
```

Where 150 is in the boot program that could be changed by a bilingual Qler and given to others who speak that language. But 1220 is in a compiled program that would not have to changed. The parentheses in 1220 keep Enter_Age$ from being mistaken for an input variable. The original programmer does not have to worry about foreign languages at all except to put a limit on the length so that changes will still fit. Yet programs could easily be adapted to any language. Of course, this doesn't solve the problem of translating user manuals.

ALTER

Once a name has been SET, it is in the name table and acts like a function. You can see it there if you type EXTRAS (requires TK2). However, SET can not be used to change a SET name; nor can it even be SET to the same thing again. Thus, if you run MIDIPlayer_boot once, and then run it again, the boot will stop. Since the boot contains LRESPR and other things that should only be done once, this is not a big problem and provision is made to start the main program directly. However, in other applications, SET names need to be changed. The ALTER keyword does just that.

Two SETs

With such a neat solution, I wrote the boot for MIDIPlayer and ran it. It did not work! It turns out that Simon's DIY SET will not work with SMSQ/E. In the nick of time, I found out about a revised SET written by George Gwilt. He had sent it in to QL Today and Jochen kindly forwarded it to me. The revised SET works just fine with SMSQ/E, however, it did not work with QDOS nor Minerva!
My solution was to rename the two SET files and place an IF in the MIDIPlayer boot:

```
If VER$='HBA' THEN
LRESPR SMS_SET_bin
ELSE
LRESPR DIY_SET_code
END IF
```

So far this solution has handled all systems that have been tried.

Do you Do DO for variety?

Many programs have the capability of adding options after the EX filename. While very useful, they are generally NOT very user friendly. That is, I find it is (usually) within my ability to figure them out, but it takes several rereads of the manual and some testing. So once I got it, I don't want to mess with it again.
But suppose in some situations, I want to run the program using different options. O No, back to the manual? No. instead I write different options (with unnumbered lines) and save them in different files with names that helps me remember what they do. I keep these files in a directory named boots. And I usually keep this directory as my default program directory:

```
PROG_USE WIN1_boots_
```

Then when I want to start a program one way, I just type DO and its name and it starts. For example, QD98, the PE editor, can be started with an option string to change a lot of the defaults: size of screen, margins, its job name, 23 things in all. To keep things within my capacity to explain, I'll just use one - which file extension (the \E switch) to be used to display lists of files. Thus, if I start QD98 (PE needs to be loaded) with:

```
EXEP win1_QD; '\E _bas'
```

All files with _bas at the end will be listed unless I change it while in the program. But if I start with:

```
EXEP win1_QD; '\E _doc'
```

then all the files with _doc at the end will be listed. Thus, if I take the first start command and save it as QDbas and the second as QDdoc in my boots directory, then all need to type is:

```
DO QDbas
```

and QD98 is started up listing _bas files. But if I type:

```
DO QDdoc
```

then it is started up listing _doc files. Maybe I could remember the file extension option, but for 5 or 10 options, I need to DO it.

This easy way did backfire on me at the recent NESQLUG meeting in New Hampshire. I did not bring the QXL which has a hard disk with the boots with me. I needed to run QLiberator, I had the QLiberator disk with me but no manual. I could not remember how to start it since at home all I type is DO Qlib and off it goes. Ah ha, I did bring backup disks of the boots and after a half hour was able to QLiberate my latest version of MIDIPlayer.

I do use ALTKEY and HOTKEY for starting some things, but there are just not enough letters in the alphabet to cover all the options I need.

To Sum Up

Using DO to start a variety of options is fine for programs that are already written and I have no control over. But when I am doing the writing, I want to make things easy for a variety of users (including myself!). But writing config programs does take time and is one more thing that can go wrong. The SET names looks like a good solution. It is easy for the user to understand and change. It will work with both compiled and interpreted programs. Plus it is a snap to program. God bless, Al.

■

QPC2 Impressions

by Don Atkins

Don Atkins writes from Australia of his impressions of the new QPC2 and its manual.

Interpreting the User's Manual

In an article I had published in the October '97 issue of Quanta Magazine (Needs of the Non-Expert) I took to task the authors of manuals over their tendency to credit far too much detailed knowledge of matters computer to purchasers of their software. I'd like to follow up on that theme with regard to the manual entitled SMSQ/E for QPC which is supplied with the new QPC2 operating system.

I mean no disrespect to the author(s) of this manual but I think there is room for improvement for those users who are not what might be termed computer professionals but who want to be able to install the software with minimum frustration - after all, there is more than enough of that in everyday life.

For QPC2 (as opposed to QPC1, the original version), there is just one short paragraph entitled "QPC2 Installation" which reads as follows:

"QPC2 does not require any special installation procedure. Run it from anywhere you want by double clicking the QPC2 text or icon. Make sure the file SMSQE.BIN is in the same directory. You have to use the SMSQE.BIN which came with QPC2, older versions are not compatible with QPC2. If QPC2 reports that it has not found 'DINPUT.DLL' then you have not installed DirectX."

Note: The file 'DirectX' is apparently part of WIN98 and was already present on my machine, presumably loaded when WIN98 was installed. There is no need to worry about SMSQE.BIN if you simply follow the QPC2 installation prompts because it will automatically be placed in the correct directory.

Whilst the paragraph tells the user how to START QPC2 I just don't think it instructs how to INSTALL it! At the risk of bringing down the fires of hell upon myself I'd like to illustrate how I would write the instructions, so here goes.

-1- Start WIN98 and choose 'My Computer'. Place the QPC2 disk in the 'A' drive and double-click on its icon. Seven files are shown, one of which is 'Install.exe' beneath a computer icon. Double-click on this to start the installation procedure. Follow the screen prompts.

-2- Unless you have good reason to alter it, accept the suggested directory location for the QPC2 files, namely C\:PROGRAM FILES\QPC2.

-3- You are asked if you want a shortcut to the Start menu. Answer Yes or No.

-4- You are next asked in which of the start menu groups you would like the QPC2 shortcut added, e.g. Accessories, Applications etc. Make your choice.

-5- You now choose whether or not to have a QPC2 shortcut added to the desktop. Most users will want to do so for convenience.

-6- Installation of QPC2 now proceeds and you will see the new icon on the desktop upon completion.

-7- Double click on the icon and the Configure program screen is displayed enabling a number of choices to be made.

-8- Choose your preferred screen display-size and the maximum resolution.

-9- At 'WIN/FLP' choose the drive devices and whether to boot from drive 'A' if a disk is present or from one of the WIN devices if there is no disk.

-10- At 'SER/PAR' select the COM ports to be used for SER1 and SER2 and the LPT port to be used for the parallel port.

-11- Under the 'General' heading allocate the amount of memory for QPC2.

-12- Choose 'ALT' or 'CTRL' for the ALT GR key. *

-13- Select the appropriate country code from the 4 offered.

-14- Choose the 'Save' button to store your preferences and check the 'Don't Show This Dialogue' button to avoid seeing the configure screen each time you start QPC2.

-15- Choose the OK button to start QPC2.

-16- Switch between QPC2 and the Win 98 desktop with the ALT & TAB key combination. "

* This GR key choice has me completely stymied - I've never come across GR before and haven't the least idea what it means - another example of unwarranted assumptions being made by an author. Even if I am missing something I should know about I'm unlikely to be the only one and my powers of logic were not up to solving this one. Somebody please enlighten us!

[The ALT GR key on European keyboards is a key to access certain extra characters printed on some keys in grey on some types of keyboard, older ones I believe. The name ALT GR stands for Alternative Graphics. ALT GR is another ALT key as far as English QL users are concerned, but some German users prefer ALT GR to simulate CTRL instead - Editor]

So that's how I would have tackled the job of instructing new users. I would welcome readers' comments and this item end any responses should assist the editors' desire to help fill up the pages of the excellent QL Today.

First Impressions of QPC2

I've been using QPC extensively since it first became available and have been waiting for its improved version, QPC2, ever since I first heard it was being developed so that it would no longer be necessary to kill off WIN98 in order to use SMSQ/E. Now it has happened, thankfully.

Firstly, the '2' version is simple to install and this is done along the lines of any other piece of software for WIN98 by activating the Install.exe file on the floppy disk supplied.

Secondly, configuring is child's play by comparison with the old QPC and there are more options provided, especially with regard to the screen display. A configuring screen is included from which to select one's choices and those choices can be saved so that they rule until you choose to alter them. There is the option of avoiding display of this screen when next starting QPC, after configuring has been done, but it is made readily available whenever it's wanted by holding down the Shift key when starting QPC2.

QPC2 can be placed in any chosen program group such as Accessories, Applications or whatever. The Installation program puts a QPC2 shortcut icon on the desktop if you choose to let it.

From then on you switch between QPC2 and your other programs in WIN98 by using the Alt+Tab key combination, or select QPC2 from its logo on the taskbar at the bottom of the screen, extremely convenient.

I left the older QPC installed on my system after installing QPC2 in case there should be any problems with using the new version.

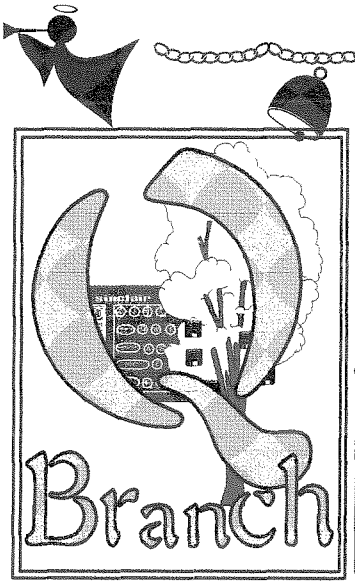
I noticed that the Win98 screen blanking facility did not work if I left QPC2 as the chosen application as it had done when using the original version of QPC. I don't understand why this should be but perhaps someone has an explanation for this? The blanking is still operative whenever the desktop is displayed.

I found a problem, so far unsolved, when I tried to uninstall QPC2. An error message said 'Uninstall log not found or corrupt. Aborting' and I got the same result when using the original disk or the file on the 'C' drive. Looks like QPC2 is here to stay .

In summary, from my first viewing of it, QPC2 is a great improvement over its predecessor and well worth investing in.

■

Q Branch



Shop hours :
Mon - Fri
10am - 6pm
Wed / Sat
10am - 2pm

Text 87

£ 79.00

Typset 94 £ 29.00
Fountext 94 £ 39.00
2488 drivers £ 29.00
Epson ESC/P2 drivers £ 26.00

Text 87 is the only QDOS / SMSQ wordprocessor capable of handliing the full screen on the Aurora / QXL / QPC systems. New drivers are currently being written.

UTILITIES

FiFi 2	£ 22.00
QSup	£ 32.00
QSpread v1.44	£ 66.00
Cueshell 2	£ 30.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 20.00
QPAC 2	£ 40.00
QTYP 2	£ 30.00
QLQ	£ 32.00

SEASONS GREETINGS FROM Q BRANCH

Well here we are at the end of the century and we seem to finally have some colour drivers to look at. After a long and patient wait the first test versions of these drivers are now running on my Q 40 in stunning technicolour. This has been a hard task for one man, and he has achieved it with a great deal of skill. The drivers are not quite ready for a general release but the ones that I am testing look very good indeed. Come along to a workshop and see them for yourself.

We expect to release a new version of the Knight Safe very soon now. Mark has been putting a lot of work into this project and the next version should be simpler to use but also have new features.

Finally we would like to wish all of our customers a Merry Christmas and a Happy and colourful New Year. We intend to continue into the next year / century / millennium and keep on providing you with the best software and hardware we can.

'Just Words' by Geoff Wicks

THESAURUS, STYLE CHECK, SOLVITPLUS 3

£ 10.00 ea / ANY 2 PROGRAMS £ 18.00 / ALL 3 PROGRAMS £ 25.00
(Includes Pointer and non-pointer driven versions)
(P.E. versions need Hot_rext, WMAN and PTR_GEN or SMSQ/E to run)
Upgrades from previous versions £ 2.50 + S.A.E. New Manuals £ 1.50

QL2PC - New program !!

Convert text files from QL to PC formats and much more !

Only £ 10.00

Spelling Crib : PD program £ 1.50 +SAE
or Free if you buy all three programs

New IBM branded ED disks !

Supplied in a sealed plastic disk box - 10 disks with labels.
Buy now while stocks last !

only £ 15.00

PROGRAMMING

QD 98	£ 53.00
QD + QBasic	£ 69.00
QD + Qliberator + QBasic	£ 110.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 37.00
Easyptr pt 1 & 2 (together)	£ 33.00
Easyptr pt 3 (C library)	£ 16.00
QMake	£ 18.00
QMon / JMon	£ 22.00
Basic Linker	£ 22.00
DISA 3	£ 37.00
QMenu	£ 16.00

Youv'e been asking for it and here it is !

The SBASIC / SuperBASIC Reference Manual

The complete definitive guide to BASIC programming in QDOS / SMSQ including three disks of PD toolkits, example procedures and an electronic index.
compiled by Rich Mellor, Franz Hermann and Peter Jaeger

Over 500 pages !
£ 40.00
+ postage

Q Branch

Feeling out on a limb ?
Reach out for Q Branch.
Suppliers of Quality QDOS/SMSQ products
Hardware and Software.

Tel +44 (0) 1903-200212 fax +44 (0) 1903-208070
email qbranch@qbranch.demon.co.uk web : http://www.qbranch.demon.co.uk

The Bank Volt, 6 Coronation Buildings
Ham Road, Worthing.
W. Sussex. BN11 2NN. UK.

Q Branch

Hardware

We have a small stock of second user items. Auroras / Qubides / Gold Cards / Qplanes / superHermes etc. call us to get details of the items available. These are going fast so call soon.

Recycled superHermes	£ 80.00 *
Recycled Gold Card	£ 60.00 *
Recycled Aurora	£ 75.00 *
Aurora	£ 90.00
Qubide	£ 55.00
Qplane	£ 25.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
'Arfa Braquet'	£ 8.00
'Son of Braquet'	£ 18.00
The 'Braquet'	£ 16.00
MC plate	£ 6.50

* when available.

14" and 15" monitors for the Aurora - Call.

ProWesS

ProWesS	£ 48.00
DATAdesign	£ 24.00
Fontutils	£ 30.00
File Search	£ 12.00
PFlist	£ 12.00
Fontpack	£ 60.00
LINEdesign v 2.16	£ 24.00
PWfile	£ 18.00

Paragraph

The ProWesS word processor

¶ Demo version £ 1.50 + postage

Full Registered version £ 18.00

Q Branch Programs

The Knight Safe 2 - with compression	£ 35.00
Q - Count Pointer driven home accounting	£ 25.00
Q - Route v1.07 Route finding programme	£ 25.00

The Fractal Collection !

This is a brand new program which will produce stunning animated fractal patterns. It will run on anything from a Gold Card to the Q40 and will be capable of using the power of the colour drivers when they are released. Complete with many example files and routines to design your own screens.

Only £ 35.00

Coming soon : The Knight Safe 3

SMSQ/E

Gold Card / Atari / QXL Version

£ 76.00

Various Atari versions : call for details

The Colour Drivers are running on the Q Branch Q 40 !
Coming to a computer near you in 2000!

QPC 2 is here !

£ 90.00 (£ 70.00 SMSQ/E Owners)

Upgrades from QPC 1 £ 30.00
(return master disk)

Special offer !
Get Cueshell for only £15.00 with any copy of SMSQ/E

There are so many extras available for the Q 40 that it is hard to list them here. If you are interested in one of these boards than contact us and we will supply the details. the basic prices are shown here.

Q 40 board with 16Mb RAM & I/O card	£ 330.00
+ SMSQ/E	£ 30.00
Extra 16 Mb RAM	£ 30.00
Tower Case	£ 45.00

All prices do not include shipping.

We can accept payment by VISA, Mastercard and Switch. You can also pay by Eurocheques made out in Sterling or a Sterling cheque drawn on a UK Bank. Prices include Post and Packing in Europe.

MIDI Sounds

Al Boehm

The program MIDIPlayer reads Standard MIDI Files (SMF) and plays them with Simon Goodwin's DIY_MIDI keywords via the QL NET port. When Simon first developed the keywords. we first had to find out whether they would work at all. They did. The next question was whether they could handle a complex song, a full orchestra for example. MIDIPlayer proved they can! MIDIPlayer has been tested with three beta versions. On 15 November, Simon's DIY_MIDI keywords and MIDIPlayer will become freeware. They will be available on the public section of the NESQLUG web page:

www.airnet.net/boehm

[Note: Hiwaay has bought Airnet. In a few months the NESQLUG web will probably be accessed by www.hiwaay.net/boehm]

The source code (SuperBasic) for MIDIPlayer will also be available as freeware so that anyone can use any of the procedures - even in commercial programs. The only caveat being that if you change a procedure, you must give the new procedure a new name so that there will not be the chaos of things with the same name doing different things. (I have this nightmare of trying to fix a bug in a program that someone else has changed.)

So far the DIY_MIDI keywords have worked on a Trump card, Gold card, SuperGold card, PCML RAM+disk, CST RAM+, and SIMPEX 256K. Hermes, Super Hermes, Minerva, SMS, Aurora, Qubide, Speedscreen, etc. have been found to make no difference. The DIY_MIDI keywords will not work on an unexpended 128 QL; it is just too slow. (Maybe they would work if placed on an EPROM.) Also the DIY_MIDI keywords will not work on the QXL. The QXL NET hardware is different. However, a QXL version is a possibility.

Hopefully, someone will write an independent review of MIDIPlayer so descriptions here will be brief. MIDIPlayer consists of two parts.

The first, MIDIPlayer_boot is in SuperBasic and sets defaults and LRESPRs the needed keywords. Such things as INK color and size of type are specified in this boot.

For the second part, you have the choice of using MIDIPlayer_bas or MIDIPlayer_obj which is the QLiberated compiled version of MIDIPlayer_bas. (I am working on a Turbo compiled version.) While the compiled _obj version does run faster, the _bas version does very good running in interpreted SuperBasic which to me was a joyful surprise.

The crux of matter is speed and timing. Once Simon had proved that the QL NET port could actually transmit a MIDI code, the question was how much music could the QL handle before it was overwhelmed.

Consider a moderate fast tempo of 120 beats per minute with a quarter note getting a beat A quarter note would require a start event, then a stop event 1/2 second later. That is, 500 milliseconds later. No problem. A 64th note would need to send a start and a stop after a 31 millisecond delay. The QL can handle it, even a chord of three 64th notes played together. But it is at this point that speed is not the main limitation. The QL can resolve time only down to a video frame which is 1/50 (1/60 on US QLs) of a second = 20 milliseconds. Thus, while the QL can handle more notes, it can resolve them properly only down to about a 16th note in a fast tempo. This time resolution is the same for a basic QL, a SGC, or a QXL whether in a compiled program or one running in interpreted SuperBasic. Thus somewhere at about the 32nd note range, notes or rests will not be evenly spaced. Can you hear it?

After listening to a lot of fast MIDI music and doubling and even quadrupling the speed, it was hard for me to detect uneven notes or rests. Indeed some of the high end MIDI sequencers add a little randomness so the notes won't sound mechanical. We get it automatically with the QL!

Instead a different phenomenon occurs - missing notes. What happens is the start and stop of a note ends up in the same time interval and the note is stopped before it even has a chance to play! Further, notes are not the only thing sent by MIDI. Things like tempo changes, e.g. retardando, tend to be no problem even though they take up compute time. What is stressful is small fast changes to notes to give character. These include small pitch changes and quick changes in volume. These do make a song interesting. I have a MIDI "Proud Mary" played with a jazz guitar imitating hammers (a guitar term when one note is quickly changed to another by pressing on the string.), slides, and notes purposefully flat to sound funky. I am pleased to report that MIDIPlayer handled it pretty well even when the song was speeded up.

Nevertheless there is a point where the number of MIDI events is too great for the QL or any other computer for that matter. The MIDI spec. gives priority to lower channels. Which I assume means if you can't play both notes, play the one on the lower channel. MIDIPlayer uses a slightly different strategy - I call it beginner timing. When a new music student gets to a hard part, he slows down to get all the notes then speeds up during the easy part. MIDIPlayer sends all the notes as fast as it can (some are lost as discussed above) and when it is caught up plays normal tempo. Keep in mind, I am using tempos far faster than usual in order to see if I could break MIDIPlayer.

I have a MIDI "Flight Of The Bumble Bee", a good test to see if a MIDI can keep up. Overall the MIDIPlayer renditions was very good. I did detect five missing notes. I am not positive, but debugging indicated these were faults of the MIDI file not of MIDIPlayer!

I have written two other MIDI programs. MIDIview_bas reads a Standard MIDI File and prints out when and what the events are. This is mainly to assist in programming or perhaps for someone who is curious to see what a Standard Midi File looks like. Another program, MIDImapper, tests the sounds of a synth that is not to the General MIDI standard and assists in finding the closest sound to a General MIDI sound. This assigning of one instrument number to another is called a MIDI map. MIDIMapper produces a file which MIDIPlayer can use. These programs can be obtained from the NESQLUG web site.

I have grandiose plans about additional MIDI programs - a MIDI editor, a notation program, a real time entry program, and more. Since I doubt I will get to writing all these programs in any reasonable time, I hope others will join me, perhaps in a joint effort or on their own, in writing QL MIDI programs. So far I have heard that Bill McKelvey, who sets up sound equipment for shows, is developing a MIDI control of amps and equalizers. Ed Kingsley wants to develop an interval tutor for voice training. And Herb Schaaf is thinking about adding sound to his fantastic mathematical displays. It should be interesting and could be downright awesome!

QBOX-USA BBS

Operating since 1993 on a Sinclair QL from Utica, Michigan, USA
Supporting *ALL* Sinclair and Timex users
Message and File Areas for QL, Z88, Spectrum, TS2068, ZX81, TS1000
Modem speeds 300 bps to 33.6k bps supported
24 hour operation - call anytime

810-254-9878

Byfleet Quanta Workshop

Dilwyn Jones

With the Q40's presence and the availability of early versions of the "colour drivers" and a lot of other good news for the QL scene, interest in this workshop was high. Dilwyn Jones reports from a *colourful* Quanta workshop.

Any mention of "colour drivers" at this workshop was guaranteed to gather a crowd of interested onlookers. The talk given by Roy Wood and Tony Firshman, with Keith Mitchell also assisting, proved to be one of the most popular sessions of the day. The colour drivers for the Q40 were far enough advanced to allow them to show some very impressive colour photographs and patterned displays generated from SBASIC. The enhanced INK and PAPER commands allowed them to specify 32 bit parameter values to make full use of the 65536 colours available under the 16-bit colour system of the Q40. Each of the RGB values could be separately set to allow a wide range of hues per colour component. As 32 bit values were used, the enhanced colours were specified using the top three bytes of the colour values (e.g. INK $FF00FF00 using the $ hex indicator made it easier to see what you were doing if you were familiar with hexadecimal notation). Presumably, if you specified the 'old' colour values 0-255 in the least significant (lowest, or rightmost byte in the hex notation system) these allowed a degree of compatibility with the old colour numbering system. Commands existed such as COLOUR to set the colour modes.

Bruce Nicholls discusses SERNET with Bernd Reinhardt

By now, the colour drivers could just about manage MODE 8 as well, unlike earlier demonstrated versions, so progress is being made sure enough. The Q40 will definitely be the first platform to get the 'colour driver' treatment, with QXL probably being second (given that QLCF club of France have made a payment to Tony Tebby towards ensuring a QXL version). The Aurora drivers will follow, and probably a new release of SMSQ/E for the Milan computer touted for ages by Jochen Merz Software. QPC will get the 'colour driver' treatment from Marcel Kilgus too, once the system is complete and sources available.

Colour drivers are released as a component of SMSQ/E. They will probably not be available for QDOS at all. The 'modular' tructure of SMSQ/E allows different sets of drivers or modules to be incorporated in the operating system for the target platform in question, making it a fairly routine matter to produce different versions of SMSQ/E for different machines.

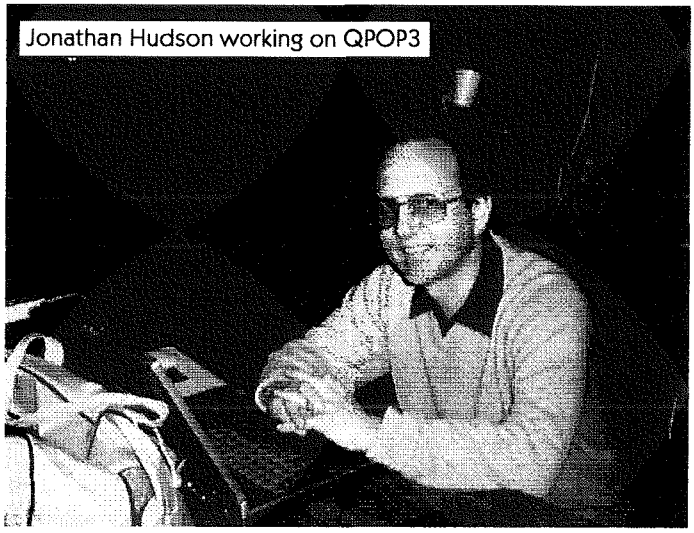
As if the colour driver and Q40 wasn't enough, Mark Knight gave a separate demonstration of his Fractal Collection on the Q40 which drew admiring gasps from everyone who saw the impressive demonstration of animated fractals using routines from Mark's Animation Toolkit. Mark has developed original fractal plotting routines and some novel ways of speeding up calculation and plotting. Mark's girlfriend works at an institute dealing with fractal research and is said to have taken some of Mark's software techniques for development there - as has often been the case, the QL leads the way! The Fractal Collection is available from QBranch.

Mark's QL software colleague David Gilham, who programmed most of the re-released Turbo Toolkit on a recent QL Today cover disk and has done some work on enhancing the Perfection word processor, was also present and able to discuss his work on the software formerly sold by Digital Precision Ltd.

Geoff Wicks of Just Words released his new QL2PC Transfer program at the show, which assists users to port text files from QL to PC and preserve formatting information etc by using the Rich Text (RTF) format accepted by most word processors on the PC, for example. QL software such as Textidy has already been available to do this task, but generally only to move files as unformatted plain text, or as QL Quill to PC Quill formats, or less used formats such as old Wordstar format, so the release of QL2PC Transfer is a welcome step forward.

Bernd Reinhardt, author of the SERNET serial network link software often bundled free with SMSQ/E for example, had made the trip over from Germany and planned to set up a corner to assist users (like myself) who'd never been able to get SERNET working properly. Unfortunately, a combination of a bug in the versio 2.24 shown at the show and the fact that nobody seemed to have a suitable serial cable meant that this exercise fell flat on its face. After the show, v2.25 was quickly released and it was found that between 2 PCs, a Laplink cable could work after all, despite protestations at the show that it may not work.

Jochen Merz Sofware has recently launched QPC2, a more Windows orientated version of its QPC emulator. Interest in this was high and Jochen is clearly on to a winner here. In time it'll be updated to work with the "colour drivers" - QPC2 is essentially SMSQ/E running on a PC as far as the user is concerned.


Jonathan Hudson working on QPOP3

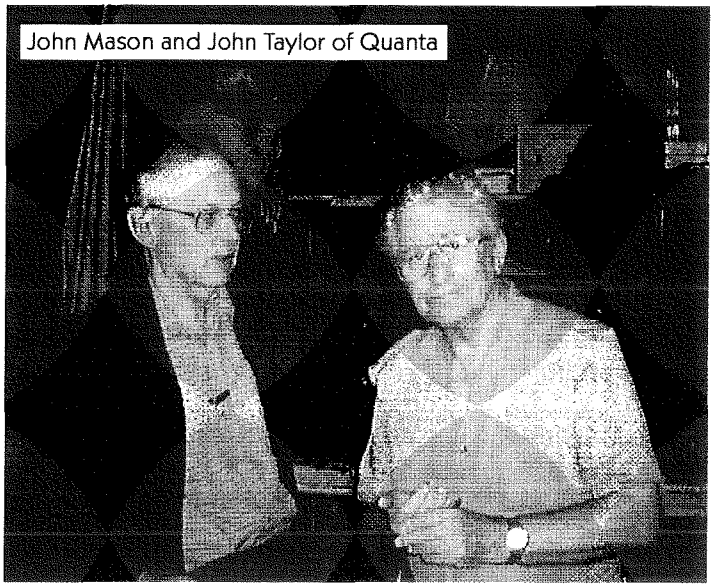
Jonathan Hudson was sat in the centre of the hall dividing his time between fielding questions about QL comms matters generally and developing his Q-POP3 software, a POP3 email client for QDOS type systems. Jonathan Hudson was aware of the work on TCP/IP systems being done in Switzerland by Jonathan Dent and was hoping that his software would in time work with that, although for now Q-POP3 will only work with platforms such as the uQLx emulator running on Linux systems giving access to the underlying comms facilities. Jonathan Hudson has ported a browser to QL systems (a version of Lynx) and some email software, so once the Swiss work on TCP/IP for the QL is complete, we can look forward to being able to send and receive emails from our QL systems as well as possibly browsing the Web. Jonathan Dent (a networking professional by trade) was quoted at the Paris show in September as saying that much of the necessary 'low level' work has been done already, so the finished system may not be that far away.

Dave Walker, famous on the QL scene for his work on the C68 compiler, was also present. He is making good progress in completing his work on the RLL (Runtime Link Library) system, which Jonathan Hudson for one was looking forward to having. This system will make life a lot easier for C programmers to produce code libraries for their software.

Q-Celt Computing had come over from Ireland for the show and launched the first in a series of CD-ROMs for QL compatible systems able to access CD-ROM drives. For now, this is restricted to emulators, since Ron Dunnett of Qubbesoft P/D (who was not at the workshop) says that CD-ROM support, while technically possible with Qubide, is not in practice possible since the programmer (Phil Borman) ran out of ROM space to include the necessary access routines and chose instead to go for ATAPI removable media drive support (Iomega ZIP and LS120 drives), and to include direct sector access facilities which in theory at least provides a means of reading CD-ROMs to some degree with careful programming. Dave Walker has been approached to extend his Discover software to read files from CD-ROMs using this direct sector access facility.

The first CD-ROM released was a collection of my commercial and freeware programs, sold for £20, a considerable saving on buying the whole collection separately from Q-Celt. Darren Branagh hopes to launch further CD-ROMs including one of clipart and fonts for Line Design users, a religious collection of Bible texts and clipart and other useful religious files, a CD-ROM of classic book text files (a kind of online electronic literatur library), a freeware QL software collection and eventually a collection of QL emulators on one CD-ROM, aimed at users who are moving or have moved to other computer systems, to try to keep those users part of the QL scene and perhaps to attract former QL users back to the fold once they realise they can use their old QL software and the wonders of SuperBASIC programming on their current machines. Darren also hopes to reach a few tinkerers and hobbyists who have never used a QL before. The CD will include most or all of the current crop of software based QL emulators, including QemuLator, QLay, QPC2 demo, uQLx and the emulaors for the Amiga, together with a collection of the best free QL

software to use witht he emulators. The best news of all is that the emulators CD (because it includes free work by various authors) will have to be freeware, so anyone can copy the emulators CD-ROM as long as they observe the right of the authors concerned. This will make it easy to distribute the CD to the world outside the QL scene as anyone can give anyone a copy (as long as they have the facility to copy CD-ROMs of course) and so we can spread the QL gospel to the big wide world outside in the new millenium!


John Mason and John Taylor of Quanta

Darren says that anyone who is interested in CD-ROMs for their systems can register their interest with him and he will contact them when the CD they are interested in is available.

CD-ROMs aside, Q-Celt brought over a small collection of non-QL toys which proved popular such as a useful multi-blade tool (must have been good, Tony Firshman bought one!), a flat lamp with long life battery and bulb (useful for keeping in cars), and also had some rare second user ED disks for sale.

On the whole the show was reasonably well attended - I have seen better and seen worse, but was most definitely a success in terms of new products and enthusiasm for current developments. It looks like we'll have a very interesting few months as we take the QL into the next millenium!

■

QL-2-PC Transfer v. 2.00

Gary Norton

There is considerable competition to the QL by inexpensive PCs. In order for the QL to remain a viable alternative to the PC, the QL must be made to work in the PC world. Several products have appeared this decade to support this realisation, such as DP's Conqueror and conversion programmes such as XOver and DisCover.

What is it?

Another, newly released program, QL-2-PC Transfer, by Geoff Wicks and Just Words! will allow word processor files created on the QL to be loaded and read, with formatting intact, by a PC's word processor. It will also work to a lesser extent in reverse.

QL-2-PC translates files created in Quill, Perfection, and Text 87 to a Rich Text Format (RTF) format. The newly created files retain their bold, underline, italics (in Perfection) tabs, margins, headers, footers, paragraphs, justifications, etc. RTF files can be opened by any word processor running under Windows, Macintosh, and probably Amiga and Atari ST operating systems.

The retention of the file's formatting is a major advance in QL-PC compatibility. Previously, if a file was to be viewed on a PC, the only option was as an ASCII text file. This retained all of the words, but there was no retention of any formatting. Now, the file will look exactly the same in Microsoft Word as in Quill.

System Requirements

The programme arrives on a single DS/DD 3.5 inch floppy disk. The computer must be a QL or a QDOS/SMSQ compatible, running Toolkit 2 and the Pointer Environment. If you are not running SMSQ-(E) or QLAY, you will also need XOver, DisCover, or QL Tools. More on these later.

Getting Started

If you don't have the Pointer Environment and Toolkit 2 initiated on start-up, you'll need a boot programme to load the TK2 extensions and the PE, the serial mouse, if desired, then EXEC the programme's obj file. If these are installed on boot-up, such as from a ROMdisq, simply do an EXEC (or EX) to the obj file. There's also a version for users of QMENU.

QL-2-PC can be installed on a hard drive or on any other type of drive.

To view the translated files, any of the above-mentioned systems will suffice to view the RTF format. What if you have a

Im stillen Winkel 12 D-47169 Duisburg
Tel. 0203 502011 Fax 0203 502012
http://www.j-m-s.com/smsq/index.htm

Ink Attack

The ink offer from the last QL Today produced pretty good reactions, so I thought it is a good idea to repeat it - but with a special christmas offer: for orders which arrive before 24th of Dec. 1999, you get a discount of 5% on ink!

High-quality alternative ink cartridges for EPSON

Styius Color 600, 800, 1520	Color	DM 21,90
Styius Color 800, 1520	Black	DM 15,90
Styius Color 500	Color	DM 19,90
Styius Color 400, 500, 600, Photo	Black	DM 15,90
Styius Color 300	4-Color	DM 22,90
Styius Color Photo, 700, EX	5-Color	DM 27,90
Styius Color 440, 640	Black	DM 17,90
Styius Color 740	Black	DM 17,90
Styius Color 440, 640, 740	Color	DM 23,90
Styius 800 (old model)	Black	DM 12,90
Styius Color II, IIs, 825, 1500	Color	DM 18,90
Styius Color II, IIs, 825, 200	Black	DM 13,90

High-quality alternative ink cartridges for CANON

BJC-70, BJC-80	Black	DM 8,99
BJC-70, BJC-80	Color	DM 10,99
BJ 300, 330	Black	DM 11,99
BJC 600, 610, 620		
Black, Cyan, Magenta, Yellow	EACH	DM 7,99
BJC 800, 820, 880		
Black, Cyan, Magenta, Yellow	EACH	DM 11,99
BJC 4000, 4100, ...	Black	DM 8,99
BJC 4000, 4100, ...	Color	DM 12,99
BJC 7000 Series	Color	DM 32,99
BJC 7000 Series	Photo	DM 36,99
BJC 7000 Series	Black	DM 34,99

Other printing stuff

CD-Labels (10 pages with 3 labels) white	DM 6,99
Inkjet paper for DOUBLE-SIDED printing A4, 360 dpi, 80g, 200 sheets	DM 12,99
Inkjet paper for DOUBLE-SIDED printing A4, 720 dpi, 250g, 50 sheets	DM 22,99
Inkjet paper single-sided printing A4, 1440 dpi, 105g, 50 sheets	DM 17,99
Inkjet glossy paper A4, 1440 dpi, 10 sheets	DM 14,99
same as above, but 50 sheets	DM 49,99
Inkjet photo paper A4, 1440 dpi, 160g, 10 sheets	DM 18,99
Inkjet glossy paper for DOUBLE-SIDED printing A4, 360 dpi, 135g, 100 sheets	DM 18,99
same as above, but 100g, 100 sheets	DM 18,99

Minimum order

4 Items or DM 80,-

TERMS OF PAYMENT

Postage and package **[Germany]** DM 8,99 (if total value of goods is up to DM 50,- then only DM 5,99). **[Europe]** DM 14,50 (if total value of goods is up to DM 50,– then only DM 9,50). **[Overseas]** between DM 14,50 (1 item) and DM 35,– (maximum). All prices incl. 15% V.A.T. (can be deducted for orders from non-EEC-countries). E&OE. Cheques in DM, £'s, Eurocheques and Credit Cards accepted.

EUROCARD. VISA MasterCard

Pre-Windows PC? That's also taken care of. There's an option to transfer to a WordPerfect v 4.2, an MS-DOS format. However, you must have a compatible word processor.

Programme Options and Operation

If you're a user of Quill, including Xchange, Perfection and/or Text 87, you're a candidate for using this programme. With Quill and Perfection, the structural formatting as well as highlighting formatting, including bold and underline. Italic is supported with Perfection. Text 87 can transfer only one type of highlighting format, but you get to choose between bold, underline, or italics. This is because of the way Text 87 stores its information.

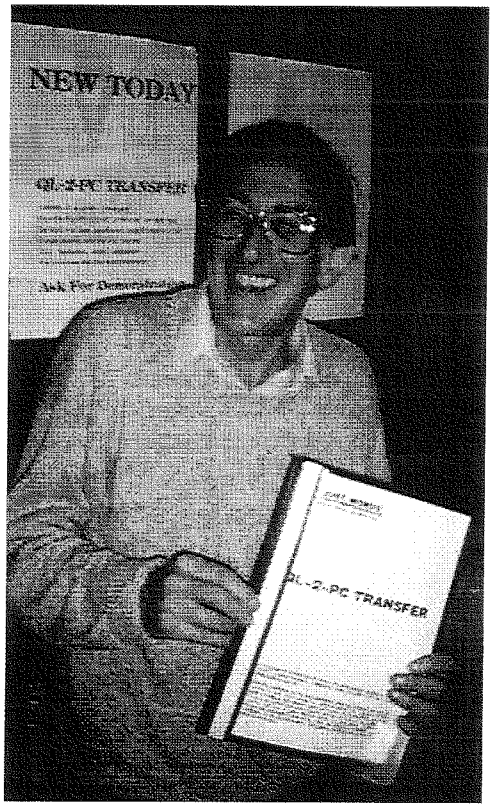
One limitation of this programme is the inability to transfer all types of highlighting formats, such as Quill's subscript and superscript. This could pose a problem in translating someone's scientific thesis or favourite recipes. H_2O or 150°C will come out as H2O and 150oC.

Using QL-2-PC

Once loaded, you're met with a well laid-out pointer environment main menu - with a fountain pen nib for a pointer - a nice touch. The main menu includes buttons to quit, information, giving the highlights of the programme, a Zzz sleep button for use with Qpac2, and a move button for use with high resolution screens. Next, there are selections for the transfer type, load document; the file to be transferred, and save to:; the output file. An RTF Format button brings you to its screen, and finally an OK button to press when you're ready to transfer.

There are several transfer options available to select. The one most people will select is the QL to PC (Windows - RTF). Other options include: QL to PC (Windows - ASCII), QL to PC (DOS - ASCII), QL to PC (DOS - WordPerfect), PC (DOS) to QL (ASCII) and PC (Windows) to QL (ASCII), E-mail message reader, and OCR Text tidy.

The difference between the Windows and DOS ASCII is the former will allow the transfer of non-English accented characters.



The e-mail and OCR options were added late in the development cycle and have only limited testing. The e-mail reader attempts to transfer e-mails while retaining its proper format. OCR text tidy attempts to remove extraneous spaces, line feeds, etc. I haven't tested these, so cannot comment on their operation.

When the RTF Format button is pressed, you come to the RTF format menu. Here you're able to set desired output font type and size, 10 types and sizes 8 through 15. You're also able to set the top & bottom margins (in cm), tab settings, and justification - left, right, left and right, centre. Finally, for Text 87 files, you choose if you want highlights for bold, underline, italic, or none. You can change any of the settings, use the default settings, save settings for specific jobs and load these saved settings. I have found the default settings to be sufficient.

How's it all work?

My overall impressions are very favourable. Navigating around the screens is just point and click. You can also easily use the cursor keys. It does all that is advertised, in a very straightforward manner. Although the programme comes with a well-documented eight page A4 manual, I was able to intuitively do all the operations without first reading it. I transferred files created in all the QL word processors supported and all, with one exception, resulted in a proper transfer.

I really tried to push the programme to its limits by transferring some rather large files. Examples of tested files include a Quill document of 5 pages, a Text 87 document of 75 pages, and two Perfection documents, the Perfection Spellchecker manual at 11 pages and the Perfection manual itself at 123 pages. All did exceptionally well except the Perfection manual. The resulting translation retained the highlighting formats - bold, underline, italic, but lost the structure format. Apparently with such large files, line feeds, tabs, etc. are lost.

I tested the QL to DOS WordPerfect transfer as well as the ASCII transfers, all without a problem. The Perfection manu-

al had the same result in both RTF and WordPerfect formats. I also tried the PC to QL transfer. I tested it against an ASCII document converted to QL format without processing it through the QL-2-PC programme. What the program's processing does, as it appeared to me, was to strip the file of line feed characters. Both the processed and unprocessed files were loaded into Perfection, and that was the only difference I could see.

As said previously, when moving a file from QL format to PC format, you require a programme such as XOver or DiscOver. If you don't have one of these, I don't know how easy they are to acquire at the present time. QL Tools, a freeware programme will also accomplish a QL to PC transfer. I actually recommend QL Tools. The two former programmes were created pre- Windows 95, so they only handle 8 character file names. Also, they can only be transferred onto double density disks. You'll get error messages if you attempt to use a HD disk.

Also mentioned, if using SMSQ you don't need a conversion programme, the operating system itself allows writing to PC DOS files. One warning, if writing to disks in this manner, you have to remember to use a file name structure such as filename.rtf as the resulting document, as this is the PC's format. QLAY was also mentioned. Since this emulator saves all files as PC files, there is no need to use a conversion program. Just load the transferred file directly into the PC word processor.

Alternatives?

QL-2-PC is a very good product and well worth the price if you must transfer files between computer platforms. If you're a Perfection or Text 87 user, it is probably the only programme available. However, if you primarily use only Quill, there is a free-ware programme, written by Mark Swift, called DOC2RTF. You can download this programme from the internet.

DOC2RTF only works with Quill files and only translates them to RFT format. But it does do its job very efficiently. There is no way of going from PC file to Quill file. Instead of having the ability of selecting the font type, size, and so forth, it has no options other than the defaults. You get an RTF file that looks exactly like the Quill file. It even translates sub and superscripts, unlike QL-2-PC. Any further formatting must be done on the PC's processor. You will still need a format translating programme, as previously mentioned to convert it to PC mode. Since DOC2RTF is free, there's good reason to own both programmes. Use this for simple Quill transfers and use QL-2-PC for more format options, or if you're using one of the other QL word processors.

A couple of facilities I'd like to see added

When you're directed to Load Document, you must know what the name of the document is. I'd like to have the option of the programme being able to give a directory of the drive containing the file, and be able to just point and click the file name to have it loaded.

I would also like to have the option of having the capability of doing a simple translate, such as what is done with DOC2RTF. Just make a simple translation using the defaults of the quill file. In most cases, that would probably be adequate. You would also know that what you're going to load on the PC will look the way you created it.

In closing, I am very impressed with QL-2-PC. Just Words! did the QL community a very valuable service in offering this type of programme. It will allow QL owners to continue to use their favourite computer platform in a world dominated by Windoze!

■

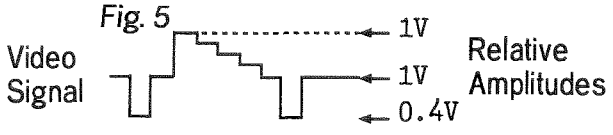
SCART Connections - Part 2

Richard Cooke

2. QL COMPOSITE SYNC to SCART

DEFINITION: Composite video/sync 1V peak to peak pin 20. See fig. 5

Fig. 5



Absolute peak to peak of sync: $CSYNC_{pp}$

$CSYNC_{pp} = (0.4*1)/(1+0.4) = 0.285$ volts

But the output from the QL is 5 Volts peak to peak and the signal is terminated in the TV by a 75 ohm load. So we add a series resistor. See Figure 6.

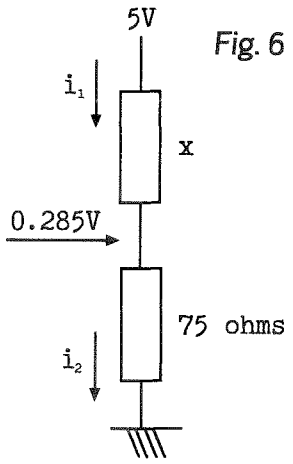


Fig. 6

We need to find the value of x.

V= IR and V/R = I

Assume current i_1 = current i_2
V/R = 0.285/75 = (5-0.285)/x = 3.8 mA
So (75*(5-0.285))/0.285 = x = 1241 ohms, about 1.2 kilohms.
The 4 band resistor colour code for which is Brown Red Red.

So what power resistor do we need?
Power = Amps * Volts
= (0.285/75) * (5-0.285) = 0.0179 watts

Reducing the current drawn by the monitor/TV increases the voltage in the QL, hence makes the Gold Card work again!

Avoiding overdriving the monitor makes the monitor work better!

Reducing the current drawn from the QL IC22 ZX8301 might reduce its chances of blowing, but at 3.8mA to ground, the current is still too high. The normal maximum is 1.8mA for TTL. See Figure 7

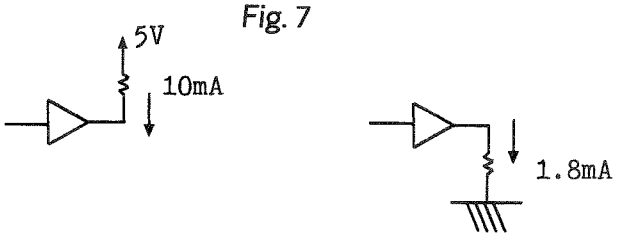


Fig. 7

If we use J7 pin 1, the signal is video, monochrome, and buffered by a transistor TR9 ZTX313. This is a cheap part to replace.

We could equally well use J7 pin 3. This is the PAL signal. Buffered by IC28 MC1377P but it is not so precise as J7 pin 1. The colour component of the signal 4MHz makes the Sync Separators work harder.

Either way you can leave out the resistor we just calculated and avoid loading the IC22 ZX8301 by taking the CSYNC signal. Use either J7 pin 1 or J7 pin 3, instead of J7 pin 5.

I attempted to use J7 pin 1 video. Terminating Red J7 pin 7, Green J7 pin 6, Blue J7 pin 8 via 330 ohm resistor into Scart RGB inputs SCART 15,9,5 caused reduced voltage at TR9 base. Vertical Sync was unstable (Picture roll)

I attempted to use J7 pin 3 PAL. This made the picture far too bright. I assume that the reduced RGB levels into IC28 MC1377P are changing the black level reference of the video signal.

BEST COMPROMISE
I return to the Brown Red Red (1K2 ohms) resistor in series with J7 pin 4 CSYNC at SCART pin 20 CVBS input.

DATA ERRORS
Note: J7 pin 5 on circuit diagram (QL Service Manual) is labelled CSYNC. This appears wrong.
J7 pin 7 labelled Green, is actually Red
J7 pin 6 labelled Red, is actually Green

Assembler part 6

Norman Dunbar

Welcome to part six of the assembler series. Don't forget that each part is now interspersed with parts of QLTdis, so you get alternating tutorial articles and articles about the project. This is a tutorial.
As you are aware from the end of part 5 of this series, we have now covered all the instructions in the 68008 processor's instruction set. Now we can start putting them to good use but in order to write good QDOS programs, you will need to know about QDOS as well - this will be covered in a later article (or two, maybe three) so we can ignore this for now.
This issue, we are going to get stuck into a failry complex part of the 68008 processor's workings - exception processing.

Oh hum - another bug

Before we proceed, I need to fix a problem in the last QLTdis article. This was pointed out by David

Gilham who actually seems to have read the article from strt to finish. He spotted that the type 28 instruction is 'MOVEA' and not 'MOVE' plus, there is no byte sized operation for this instruction. Also noted by David was the fact that there should be a byte sized version for the type 29 instruction. The actual article should read as follows:

Type twenty-eight.

The MOVEA ‹ea› An instruction is the only one in this family. The variable bits are 0 to 5 - as usual, 9 to 11 define the destination address register and bits 12 and 13 the size. The mask is $C1C0 and the result is $0040. The size is non-standard in this instruction:

2	long
3	word

Type twenty-nine.

Nearly done now! This is probably the most difficult instruction to decode as it has two effective addresses in it. There is only one instruction in this family - the MOVE ‹ea›,‹ea›. The mask is $C000 and the result is $0000. Bits 0 to 5 define the effective address of the source and bits 9 to 11 define the effective address of the destination. Bits 12 and 13 define the size as follows:

1	byte
2	long
3	word

Here is another bug. While working on the algorithms for the individual instruction type decoding, I came across a problem with type eleven instructions. I have made quite a big mistake here. The ABCD and SBCD instructions are ok, but the ADDX and SUBX ones have a size which is defined in bits 6 and 7. For this reason, they have been moved out of this family. It should now look like this:

Type eleven.

Bits 0 to 2 define a source register, bits 9 to 11 the destination register and bit 3 defines whether the instruction is operating on memory or not. This family has a mask word for each member. The instructions, masks and results are:

ABCD.size	$F1F0	$C100
(Mask then result)		
SBCD.size	$F1F0	$8100

If bit 3 is set then the address register with pre-decrement form is used :

```
ABCD.size   -(Ax),-(Ay)
SBCD.size   -(Ax),-(Ay)
```

and if not, the data register form is used:

```
ABCD   Dx,Dy
SBCD   Dx,Dy
```

Size is the usual:

0 = byte 1 = word 2 = long

Quite a tricky instruction.

This means that the ADDX and SUBX instructions need a new home. I have checked around and only the CMPM family is similar (type thirteen) but the difference was too great to distinguish between them, so a new type thirty has been created as follows:

Type thirty

Bits 0 to 2 define a source register, bits 9 to 11 the destination register, bits 6 and 7 define the size and bit 3 defines whether the instruction is operating on memory or not. This family has a mask word for each member. The instructions, masks and results are:

ADDX	$F130	$D100
(Mask then result)		
SUBX	$F130	$9100

If bit 3 is set then the address register with pre-decrement form is used:

```
ADDX   -(Ax),-(Ay)
SUBX   -(Ax),-(Ay)
```

and if not, the data register form is used:

```
ADDX   Dx,Dy
SUBX   Dx,Dy
```

This means that the old type thirty has been demoted to type thirty one:

Type thirty one

This is the catch all at the end of the list. Anything which has not been detected by this time, will always be caught here. The mask is $0000 and the result is $0000. If we catch an instruction here, the chances are that it was data and to show this, we simply define it as a constant word of data. The output is:

```
DC.W    $xxxx
```

Right, bugs over, on with exceptions.

Exceptions.

As mentioned in the instruction summary in past articles, the QL processor runs in two modes - user and supervisor - and some instructions cannot be run in user mode without causing an exception to be generated. I promised to explain what these exceptions are, so here goes ....

An exception is an event or happening that

causes the processor to deviate from its normal course of action and to jump to a predetermined place in the operating system where it starts executing a piece of code that handles such events. In QDOS (the QL's operating system) many of these routines have been 'botched' in an effort to save on memory and others simply do nothing. This is unfortunate, however, all is not lost.

All 68000 series processors have an area of memory set aside to hold the exception table. This table is 1024 bytes long and holds a full set of exception vectors - basically a long word holding the address of the sub-routine that handles the appropriate exception. In QDOS this table is only partially there as will become clear. There are 256 vectors normally, each one being 4 bytes long. Vector zero is at address zero in the memory map and vector 255 is at address $3FC.
The vector table SHOULD look like the following:

Vector	Address	Purpose
000	0000	Reset - SSP value
001	0004	Reset - USP value
002	0008	Bus Error
003	000C	Address Error
004	0010	ILLEGAL Instruction
005	0014	Divide by zero
006	0018	CHK instruction
007	001C	TRAPV instruction
008	0020	Privilege violation
009	0024	Trace
010	0028	Line 1010 emulator
011	002C	Line 1111 emulator
012	0030	Reserved for Motorola
013	0034	Reserved for Motorola
014	0038	Reserved for Motorola
015	003C	Uninitialised Interrupt
016	0040	Reserved for Motorola
:	:	: :
:	:	: :
024	0060	Spurious interrupt
025	0064	Interrupt level 1
026	0068	Interrupt level 2
027	006C	Interrupt level 3
028	0070	Interrupt level 4
029	0074	Interrupt level 5
030	0078	Interrupt level 6
031	007C	Interrupt level 7
032	0080	TRAP #0
033	0084	TRAP #1
034	0088	TRAP #2
035	008C	TRAP #3
036	0090	TRAP #4
037	0094	TRAP #5
038	0098	TRAP #6
039	009C	TRAP #7
040	00A0	TRAP #8
041	00A4	TRAP #9
042	00A8	TRAP #10
043	00AC	TRAP #11
044	00B0	TRAP #12
045	00B4	TRAP #13
046	00B8	TRAP #14
047	00BC	TRAP #15
048	00C0	Reserved for Motorola
:	:	: :
:	:	: :
064	0100	User vector 1
:	:	: :
:	:	: :
255	03FF	User vector 192

It can be seen that a huge number of the vectors are reserved for Motorola to use in future processors. The User vectors look interesting, but have been obliterated by some of the code in QDOS and cannot be used.

On the QL, the vectors are as follows:

Vec.	Addr.	Purpose
000	0000	Reset - SSP value
001	0004	Reset - USP value
002	0008	Bus Error - IGNORED
003	000C	Address Error - MAY BE REDEFINED
004	0010	ILLEGAL Instruction - MAY BE REDEF
005	0014	Divide by zero - MAY BE REDEFINED
006	0018	CHK instruction - MAY BE REDEFINED
007	001C	TRAPV instruction - MAY BE REDEF
008	0020	Privilege violation - MAY BE REDEFINED
009	0024	Trace - MAY BE REDEFINED
010	0028	Line 1010 emulator - UNUSABLE
011	002C	Line 1111 emulator - UNUSABLE
012	0030	Reserved for Motorola - UNUSABLE
013	0034	Reserved for Motorola - UNUSABLE
014	0038	Reserved for Motorola - UNUSABLE
015	003C	Uninitialised Interrupt - UNUSABLE
016	0040	Reserved for Motorola - UNUSABLE
:	:	: :
:	:	: :
024	0060	Spurious interrupt - IGNORED
025	0064	Interrupt level 1 - IGNORED
026	0068	Interrupt level 2 - QL System interrupt
027	006C	Interrupt level 3 - IGNORED
028	0070	Interrupt level 4 - IGNORED
029	0074	Interrupt level 5 - IGNORED
030	0078	Interrupt level 6 - IGNORED
031	007C	Interrupt level 7 - HANGS THE QL - MAY BE REDEFINED
032	0080	TRAP #0 - Make a call to QDOS
033	0084	TRAP #1 - Make a call to QDOS
034	0088	TRAP #2 - Make a call to QDOS
035	008C	TRAP #3 - Make a call to QDOS
036	0090	TRAP #4 - Make a call to QDOS
037	0094	TRAP #5 - IGNORED - MAY BE REDEF
038	0098	TRAP #6 - IGNORED - MAY BE REDEF

039	009C	TRAP #7 - IGNORED - MAY BE REDEF
040	00A0	TRAP #8 - IGNORED - MAY BE REDEF
041	00A4	TRAP #9 - IGNORED - MAY BE REDEF
042	00A8	TRAP #10 - IGNORED - MAY BE REDEF
043	00AC	TRAP #11 - IGNORED - MAY BE REDEF
044	00B0	TRAP #12 - IGNORED - MAY BE REDEF
045	00B4	TRAP #13 - IGNORED - MAY BE REDEF
046	00B8	TRAP #14 - IGNORED - MAY BE REDEF
047	00BC	TRAP #15 - IGNORED - MAY BE REDEF
048	00C0	Reserved for Motorola - UNUSABLE
:	:	: :
:	:	: :
064	0100	User vector 1 - UNUSABLE
:	:	: :
:	:	: :
255	03FF	User vector 192 - UNUSABLE

All vectors marked 'UNUSABLE' have been batched in the ROM and have bits of code in place of the vectors. So you can see not much is left. The designers of QDOS didn't have enough room in the early ROMs to fit all the code in - some QLs even came with a 'dongle' hanging out of the external ROM slot so that they could fit all the code in. Later versions got rid of the dongle, but the vector table had been 'redesigned' to make the code fit. Luckily they did allow a number of the exceptions to be redefined so that programmers could write their own routines to handle these exceptions.

Working QDOS Exceptions

RESET vectors 0 and 1 - these two vectors are simply the values that are put into the SSP and USP on system power up. Vector 0 gives the value for the stack pointer for supervisor mode and vector 1 gives the stack pointer for user mode.

ADDRESS ERROR - this occurs whenever the processor tries to do a word or long sized operation or access at an odd address. For example, the following code fragment will cause an address error:

```
MOVEA.L   #1,A1
MOVE.W    (A1),D0
```

On a normal QL this will usually cause the system to hang, but as the vector can be redefined, we can use it to point to an address that can correctly handle this error. More on this later.

ILLEGAL INSTRUCTION - this occurs when an instruction is executed that is not a valid instruction for the processor, or when the ILLEGAL instruction is executed. Illegal usually crash the QL, but can be handled by our own routines.

DIVIDE BY ZERO - This should be obvious. This is ignored on the QL, but can be redefined for our own use.

CHK INSTRUCTION - Called when the CHK instruction is used and the value in a data register is out of bounds, Ignored on the QL but redefinable.

TRAPV INSTRUCTION - Called when the TRAPV instruction is executed and the V flag is set. Ignored by the QL but, once again, is redefinable.

PRIVILEGE VIOLATION - When a program running in user mode attempts to execute an instruction that is privileged, this exception is raised. Ignored by the QL, but redefinable.

TRACE - If the trace (T) bit is set in the Status Register, this execption is generated after each instruction. Can be redefind to call code in a machine code monitor program, but usually ignored by the QL.

INTERRUPT LEVEL 2 - there are 7 levels of interrupt on a normal 68000 series processor, but only one is used on the QL. The level 2 interrupt is generated by the internal electronics and causes the keyboard to be scanned, the scheduler to switch tasks etc. Levels 1 and 3 to 6 are ignored on the QL.

INTERRUPT LEVEL 7 - Level 7 is the non-maskable interrupt and is raised when you press CTRL ALT 7 together. When the QL hardware was being built and debugged, some external equipment was 'bolted on' and this combination of keys caused a level 7 interrupt which activated the debugging equipment. Unfortunately, when the QL went into production, the code was left in and pressing these keys together is a pretty good way to trash the system. May be redefined for our own use - this could be fun !

TRAP #0 - Switch the QL into supervisor mode and cause the SSP version of A7 to be used.

TRAP #1 - this is the QDOS manager trap and is used to control resources in the QL such as baud rates, jobs, memory allocation and deallocation etc.

TRAP #2 - this is the QL's I/O manager trap and is used to open & close channels as well as foramtting discs and deleting files.

TRAP #3 - This allows QDOS to read data from channels, queues, set colours etc.

TRAP #4 - Used by the SuperBasic interpreter to switch between A6 relative and Absolute addresses when calling various routines.

TRAP #5 to TRAP #15 - these are unused on the QL but can be redefined.

What happens when an exception occurs?

When an exception occurs, some data is put onto the stack prior to the exception being processed. Remember, the stack pointer is the SSP and not the normal USP variant of the A7 register.

For most exceptions, the data put onto the stack is simply the program counter and the status register as follows:

```
High address -> PC low word
                PC high word
SSP ---------->  Status register word
```

so when the exception handler is running, the stack pointer holds the address of the SR at the time the exception was caused and 4(A7) holds the program counter where the exception was caused.
The above is true for all exceptions apart from BUS ERROR, ADDRESS ERROR or RESET. These three have a different stack frame:

```
High address -> PC low word
                PC high word
                Status register word
                Instruction register word
                Access address low word
                Access address high word
SSP ---------->  Access type and function
                   code (one word)
```

This additional data includes a copy of the first word of the instruction that was being processed when the exception was caused, the address that was being accessed when the exception was caused and a word describing what the processor was trying to do at the time.
Note that the value in the program counter on the stack is not always the actual address of the start of the instruction - it could be anything from the next word or even the address 10 bytes on from the actual address of the instruction - beware.

At the end of an exception processing routine an RTE instruction is used to restore the status register and the program counter from the stack. It follows then that in the case of an ADDRESS or BUS exception that this is gaing to fail unless the additional data is first cleared from the stack - or a 68020 used instead!

Building an exception handler.

I suppose we need to built an exception handler now! In the QL you build a table of vectors for the following exceptions:

Address error
Illegal
Divide by zero
CHK
TRAPV
Privilege
Trace
Interrupt level 7
Trap #5
Trap #6
Trap #7
Trap #8
Trap #9
Trap #10
Trap #11
Trap #12
Trap #13
Trap #14
Trap #15

And then tell QDOS to use this table for your job. Any exceptions that are generated and that are mentioned above will be handled by your own routine. Of all of these, the address error needs to have special treatment because it has the extra data on the stack.
The problem being that if your instruction caused an error, whet happens when you handle the exception and RTE - does the program fail again because it tried to execute the same instruction again? Sometimes is the only answer.
The following code will be very useful when you first start writing assembler as it will trap the exceptions mentioned above and attempt to allow you to carry on. This example should be run on a 68000 or 68008 ONLY. I do not have the data for exception handling on a 68020 or above so Gold Cards, Super Gold Cards etc may cause problems. I don't know.

The exception handler code.

```
*===========================================================================
* This code adds a 'protective barrier' to the QL so that silly programming
* errors can be intercepted and hopefully handled before the QL crashes out.
*
* This code should only be run on a 68000 or 68008 as the exception stack
* frame is (probably) different on 68010 and above.
*
* Copyright Normn Dunbar 1999 but permission for unlimited use and abuse is
* given !
*===========================================================================

start       lea exceptions,a1       ; Table of exceptions (empty !)
            lea x_address,a2        ; Routine for address exceptions
            move.l a2,(a1)+         ; Save in table
```

TF Services

superHermes

A major hardware upgrade for the QL

All Hermes features (see below for list) PLUS full 19200 throughput on ser1/ser2 not affected by sound IBM AT keyboard interface (plus foreign drivers) // HIGH SPEED RS232 industry standard two-way serial port. 4800cps throughput (supergoldcard - qtpi - zmodem) at 57600bps // THREE low speed RS232 inputs (1200 to 30bps) - Driver for SERIAL MOUSE supplied. Other uses include RTTY/graphics tablet etc // THREE spare I/O lines (logic) with GND/+5V // Capslock/scrollock LED connector // Turbo/keylock connectors // 1.5k user data permanently storeable in EEPROM

All this on a professional board about twice the size of the 8049 co-processor it replaces

Cost (including manual/software). **£90** (£92/£87/£90)
IBM AT UK layout Keyboard..... **£22** (£24/£23/£27)
Serial mouse ..... **£11** (£13/£12/£14)
Capslock/scrollock LED ..... **£1** (£1.50/£1/£1.50)
Keyboard or mouse lead..... **£3** (£3.50/£3/£3.50)
High speed serial (ser3) lead ..... **£4** (£4.50/£4/£4.50)

superHermes LITE

All Hermes features (see above) + an IBM AT keyboard interface only. Entry level superHermes.
Cost (incl keyboard lead)...**£53** (£55.50/£51/£53.50)

Minerva

MINERVA RTC (MKII) + battery for 256 bytes ram. CRASHPROOF clock & I²C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

DEBUGGED operating system/ autoboot on reset of power failure/ Multiple Basic/ faster scheduler- graphics (within 10% of lightning) - string handling/ WHEN ERROR/ 2nd screen/ TRACE/ non-English keyboard drivers/ "warm" fast reset. V1.97 with split OUTPUT baud rates (+ Hermes) & built in Multibasic.

First upgrade free. Otherwise send £3 (+£5 for manual if requd). Send disk plus SAE or two IRCs

MKI...£40 (£41/£40/£43) **MKII...£65** (£66/£63/£67)

QL REPAIRS (UK only)

Fixed price for unmodified QLs, excl microdrives. QLs tested with Thorn-EMI rig and ROM software.

£27 incl 6 month guarantee

QL RomDisq

Up to 8 mbyte of flash memory for the QL

A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent storage - it can be thought of as a portable hard disk on a card, and reads at some 2 mbytes per second. Think of it - you could fully boot an expanded QL, including all drivers/SMSQ etc off **RomDisq** at hard disk speed with only a memory expansion needed.

2 mbytes RomDisq..... **£39** (£41£37/£40)
4mbytes RomDisq..... **£65**(£66/£63/£67)
8 mbytes RomDisq..... **£98** (£100/£95/£99)
Aurora adaptor..... **£3** (£3.50/£3/£4)

MPLANE

A low profile powered backplane with ROM port

A three expansion backplane with ROM port included for RomDisq etc. Aurora can be fitted in notebook case and powered off single 5V rail - contact QBranch for details. Two boards (eg Aurora and Gold Card/Super Gold Card/Goldfire fixed to base. Suitable for Aurora (ROM accessible from outside) & QL motherboard in tower case. Specify ROM facing IN towards boards, or OUT towards back of case.

Cost..... **£34** (£36/£33/£35)

I2C INTERFACES

Connects to Minerva MKII and any Philips I²C bus

Power Driver Interface **16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)**
2 amp (for 8 relays, small motors)..... **£40** (£43/£38/£44)
4 amp total (for motors etc) ..... **£45** (£48/£43/£50)
Relays (8 3a 12v 2-way mains relays (needs 2a power driver)..... **£25** (£28/£23/£27)
Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required..**£25** (£28/£23/£27)
Analogue Interface **Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temp measurements, sound sampling (to 5 KHz), x/y plotting** .....**£30** (£31.50/£29/£30)
Temp probe (-40°C to +125°C)..... **£10** (£10.50/£10/£11)
Connector for four temp probes..... **£10** (£10.50/£10/£11)
Data sheets..... **£2** (£2.50/£2/£3)

QL SPARES

Keyboard membrane..... **£24** (£25/£24/£27)
1377 PAL ..... **£3** (£3.50/£3/£4)
Circuit diagrams..... **£3** (£3.50/£3/£4)
68008 cpu or 8049 IPC ..... **£8** (£8.50/£7.50/£9)
8301/8302 or JM ROM or serial lead.. **£10** (£11.50/£10/£11)
Power supply (sea mail overseas)..... **£12** (£17/£16/£21)

Prices include postage and packing (Airmail where applicable). Prices are: UK (EC,/Europe outside EC/Rest of world). Payment by cheque drawn on bank with UK address, debit card/Mastercard/Access/Eurocard/postal order or CASH! (No Eurocheques). SAE or IRC for full list and details

24 Sep 99

VISA

29 Longfield Road, TRING, Herts, HP23 4DG
Tel: 01442-828254 Fax/BBS: 01442-828255
tony@firshman.demon.co.uk http://www.firshman.demon.co.uk

MasterCard

```
            lea x_illegal,a2        ; Routine for illegal exceptions
            move.l a2,(a1)+         ; Save in table

            lea x_divide,a2         ; Divide by zero
            move.l a2,(a1)+         ; Save in table

            lea x_check,a2          ; CHK instruction
            move.l a2,(a1)+         ; Save in table

            lea x_trapv,a2          ; TRAPV instruction
            move.l a2,(a1)+         ; Save in table

            lea x_priv,a2           ; Privilege violation
            move.l a2,(a1)+         ; Save in table

            lea x_trace,a2          ; Trace exception
            move.l a2,(a1)+         ; Save in table

            lea x_int_7,a2          ; Interrupt level 7
            move.l a2,(a1)+         ; Save in table

            lea x_trap,a2           ; TRAP #5 to TRAP #15
            moveq #10,d0            ; There are 11 entries to fill

trap_loop   move.l a2,(a1)+         ; Save one in table
            dbra d0,trap_loop       ; Then do the rest

            lea exceptions,a1       ; Exceptions table again - full
            moveq #-1,d1            ; Job id = 'this job'
            moveq #mt_trapv,d0      ; Routine to set exception table
            trap #1                 ; And do it
            rts                     ; Return to SuperBasic with error code in D0

*===========================================================================
* Now the actual exception handlers themselves. Apart from the ADDRESS
* exception, all have 3 words on the stack when called. ADDRESS has more.
*===========================================================================

x_address   lea t_address,a1    ; Message to print
            bsr message_0       ; Print the message
            addq.l #8,A7        ; Tidy extra data off the stack
            rte                 ; Attempt to continue

t_address   dc.w 15
            dc.b 'ADDRESS error.'
            dc.b 10

x_illegal   lea t_illegal,a1    ; Message to print
            bsr message_0       ; Print the message
            addq.l #2,2(a7)     ; Don't execute this instruction again !
            rte                 ; Attempt the next instruction

t_illegal   dc.w 21
            dc.b 'ILLEGAL instruction.'
            dc.b 10


x_divide    lea t_divide,a1     ; Message to print
            bsr message_0       ; Print the message
            rte                 ; Attempt to carry on

t_divide    dc.w 16
            dc.b 'DIVIDE BY ZERO.'
            dc.b 10


x_check     lea t_check,a1      ; Message to print
            bsr message_0       ; Print the message
            rte                 ; Attempt to carry on

t_check     dc.w 17
            dc.b 'CHK instruction.'
            dc.b 10
```

```
x_trapv     lea t_trapv,a1      ; Message to print
            bsr message_0       ; Print the message
            rte                 ; Attempt to carry on

t_trapv     dc.w 19
            dc.b 'TRAPV instruction.'
            dc.b 10

x_priv      lea t_priv,a1       ; Message to print
            bsr message_0       ; Print the message
            rte                 ; Attempt to carry on

t_priv      dc.w 21
            dc.b 'PRIVILEGE VIOLATION.'
            dc.b 10

x_trace lea t_trace,a1          ; Message to print
            bsr message_0       ; Print the message
            rte                 ; Attempt to carry on

t_trace     dc.w 25
            dc.b 'TRACE - not implemented.'
            dc.b 10

x_int_7 lea t_int_7,a1          ; Message to print
            bsr message_0       ; Print the message
            rte                 ; Attempt to carry on

t_int_7     dc.w 26
            dc.b 'DO NOT PRESS CTRL ALT 7 !'
            dc.b 10

x_trap      lea t_trap,a1       ; Message to print
            bsr message_0       ; Print the message
            rte                 ; Attempt to carry on

t_trap      dc.w 39
            dc.b 'TRAP #5 to TRAP #15 - not implemented.'
            dc.b 10

*=============================================================================
* This routine prints a message to channel 0. The message is at 0(A1) in the
* usual QDOS format of a size word followed by the text. The UT_ERR0 routine
* expects an error code in D0 -or- the address of a user defined error message
* in D0 with bit 31 set to show that it is user defined.
*=============================================================================
message_0   move.l a1,d0        ; Address of user defined error message
            bset #31,d0         ; Mark as user defined
            move.w ut_err0,a2   ; The required routine is vectored (ut_errZERO)
            jsr (a2)            ; Do the routine & print the message
            moveq #0,d0         ; Clear error flag
            rts

exceptions  ds.l    19          ; There are 19 exceptions to define

*=============================================================================
* HERE ENDETH THE CODE.
*=============================================================================
```

How it works.

Now that you have typed the above code into a file, I shall explain what is happening. The code begins at the label 'start' and sets A1 to the address of the label 'exceptions' within the program. This is where the LEA instruction is useful - when writing position independant programs. These are programs that can be run at any address and are a requirement if you want to write good QDOS programs.

The 'exceptions' label identifies the start of the 19 long words of data that hold the addresses of the 19 redefined exception vectors as detailed above. At the moment the table contains random garbage and needs to be initialised BEFORE we tell QDOS to use the new vectors.

The address of the routine to handle address exceptions, 'x_address', is loaded into A2 - again using position independant methods, and then placed in the table at the first location. You will

note that 'address register with post-increment' addressing is used here. This means that A1 is automatically incremented by the correct amount - 4 in the case of the long sized move - ready for the next vector to be loaded.
This process is repeated for the illegal, divide by zero, CHK, TRAPV, privilege violation, trace and interrupt level 7 vectors.
There are 11 vectors left in the table for TRAP #5 through to TRAP #15. Rather than give each of these a single handler, we point them all to the same one as we intend to ignore these instructions when they occur. To set these 11 vectors up, we run through a small loop which counts D0 down from 10 to -1 setting the vector for each of the 11 TRAP exceptions to be the single routine at address x_trap.
Our exceptions table has now been defined and all we have to do is tell QDOS that we want to use it. Once again, A1 is set to the start address of the exceptions table as required by QDOS, D1 is then set to -1 which implies 'the current job' to QDOS. This is used in many of the QDOS routines which require a job ID, passing -1 means 'me'. As we are executing this code directly from SuperBasic, that is what the current job will be. Once the vectors have been set up for any job, all other jobs created by it will use the same vector table.
This means that as the initiating job is SuperBasic, and as most other jobs are created by SuperBasic, this means that we have effectively created a protection mechanism for every job in the system created FROM THIS POINT ONWARDS ! If this is the first code loaded on your system, then every single job created will be protected by this code.
Trap #1 is called with D0 set to the value MT_TRAPV - a fancy way of saying 7 - and we return to SuperBasic with any error codes that may arise. As there appears to be only 'invalid job' returned, it is unlikely that there will be any as we are using the current job's own id.
Now that the initialisation has been carried out, the exception handlers will just sit there until such time as they are activated.
Most of the handler code is the same - we simply trap the exception, print a warning message to channel #0 and attempt to carry on - but the Address and illegal exception handlers do additional processing.
In the case of an address error, there is an extra 8 bytes of data on the stack on top of the 'standard' stack frame as discussed above. These need to be cleared off before we execute the RTE instruction. **THIS IS ONLY TRUE OF A QL WITH 128K OR A TRUMP CARD ETC. IF YOU USE QXL OR SOME OTHER CARD WITH AN UPGRADED PROCESSOR, THEN THE STACK IS DIFFERENT AND THIS CODE WON'T WORK PROPERLY.**
An Illegal instruction also manipulates the stack, but this time, it adds 2 to the address of the failed instruction. This prevents it from trying to execute it again when we exit the routine. Of course this may not always be successful and can cause further errors along the way - if the instruction was followed by a word of data for example. Trying to execute the data could lead to another exception and so on. What would you rather have, a message telling you about it or a lock up with no indications?
The messages are defined in the standard QDOS manner of a size word followed by the bytes of the message. The appropriate message has its address loaded into A1 by the exception handler, and a brach is made to the sub-routine MESSAGE_0 which will attempt to display a message to channel #0. If this fails, it will try #1 before giving up.
If you have a QDOS manual and you look up UT_ERR0 (that's a zero by the way !) you will see that it takes an error code in D0 as its only parameter. We are using it slightly differently as we are defining our own messages and not using the Sinclair defined ones such as 'invalid channel id' or 'bad parameter' etc.In order to do this, we load D0 with the address of the message but set bit 31 of D0 so that QDOS knows that it is an address and not an error code.
The UT_ERR0 routine lives in the ROM somewhere, I don't know where it lives in all ROMs as it could have been moved between ROM releases. Because of this, there is a vector table in the ROM at a standard position. To get the address of the routine, we simply read the contents of the vector table into an address register and JSR to that address. (This will be explained later in the series when I cover QDOS).
So now that we have assembled the code all we do is LRESPR it (or RESPR(512), LBYTES and then CALL) and that is it. Whenever any exceptions occur, the above code will handle them and, most importantly, tell you what has happened. Your QL may still be hung - but at least you should know why!

Next time we shall be back in the guts of QLTdis with a lot more code for you to type in and also the building up of the decoding table. See you then.
■

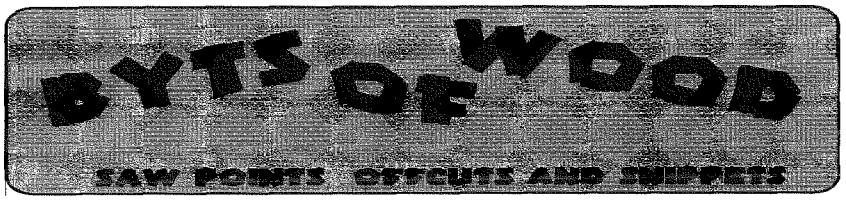
Last Minute PROGS News

It has finally happened, I seem to have produced a version of ProWesS which is Q40 compatible.
I haven't quite finished the distribution yet, but here are the changes... Ok, not too many, just a new dll manager (with better cache handling), a new version of the screen driver which detects and handles mode 33, and a mode 33 bitmap driver.
Updates should be available in the dedicated ProWesS area of the JMS BBS.

Out of Space

We are sorry that we have run out of space in the magazine to include part 2 of the Hotkeys article, which will now appear in the next issue.
The same applies to other promised articles, like the next part of the Thing article. We had to compensate a bit for the extra weight added by the bonus calendar (weights more than 4 A4 pages) and the cover disk.

Expect it in the next issue.



BBS Again.

Well I got it wrong again but at least this time it was not all my fault. I logged on to Phil Borman's Web Site to get the correct phone number for the Nene Valley BBS and that was the number that I put into my last column. Unfortunately Phil had not updated his website with the new number so the one that I published was wrong. The correct number is:
01933-389482

Brought to QBook

The above problem is not mine alone. Jonathan Hudson has just released v 1.66 of QTPI. Although there are not a lot of new features in this version it does fix one or two bugs which have crept in or been reported.
Included in the package are the example files which include QBOOK_DIST, the phone book used by QTPI to make the calls. Both Phil's number and Tony Firshman's number in this are wrong and, for some obscure reason, Jonathan does not seem to know that he moved from the Middle East last year and still includes his own 'Dead Letter Drop' number in it.
You can find an updated copy of the QBOOK for QTPI on T.F. Services Website and on their BBS. (see their ad for the number)

All Singing All Dancing Display

Both my Aurora systems developed a fault this month. First to go was my trusty old tower case system which insisted on displaying a flickering version of the lower part of the screen across the centre for a few minutes after being turned on. this cleared after a while and the system remained stable until it was turned off and turned on again. Next my MinisQL started to play up. This was a different fault in that parts of the display kept getting left behind and not redrawn, the buttons were a bit distorted although the system could be still be used with a bit of perseverance.
The MinisQL was the most important thing to fix because it is not only used for the Q Branch stuff but also holds all of the accounts for the shop. Although I am a good boy and had them all backed up onto both the Tower cased system and the two QPC 2 systems it would be very inconvenient to be without this machine for any length of time. Out came the screwdriver.
Looking inside I could see little out of place so I removed the 'SCAN' chip from the Aurora board and replaced it with a spare for another Aurora. A restart quickly confirmed that this had not solved the problem. Next I recalled that a constant source of problems with my old QLs was the 8302 chip. These often had corroded legs for some reason so I removed that and lo and behold they were in a very bad state. I applied Tony Firshman's pencil eraser solution and replaced the chip. All was fine. I then went home and tackled the Tower case. That was suffering from the same problem.
If you have any display problems these are the chips to look at first.

Great Leaping Mice!

Another problem which has been causing me extended grief on the MinisQL is the behaviour of my little mouse on the MinisQL. Ever since we built it the silly thing has persisted in leaping around the screen at totally the wrong moments -

ruining my score in The Lonely Joker and making drawing using LINEdesign very hard indeed.

Since my MinisQL was prototype 2 and had a backplane made from a converted Falkenberg model cut down to fit and a superHermes which had its connections soldered onto the board because there was no space for extra plugs I was pretty sure that the best thing to do was to rebuild the whole thing using one of T.F. Services superb Mplanes.

Keith Mitchell did this for me and, when the whole thing was reassembled the system felt a lot better to use but the mouse still leapt around. Keith then removed the old, modified, superHermes and replaced it with a full version. I was then able to stop using the Albin Hessler SERmouse software and go over to the superHermes one. The mouse is now rock steady and the whole system is a joy to use.

This is no criticism of Albin's software. I suspect that the problem lies in the power supply. The superHermes mouse could be less sensitive to lower voltages on the serial port especially since the MinisQl is powered with only a 5 volt supply and the 12 volt supply is generated by a chip on the Mplane.

I now use the Mplane on both of my Aurora systems and I think it is a better unit than the old Qplane. The noise produced by the Super Gold Card seems less noticeable and the whole arrangement makes for a much more stable assembly in either a Tower Case or a MinisQl.

Dog Days

Computers, like dogs and cats, have a different timescale to humans. When you trusty old hound is barely into his teens he is, in dog terms, firmly in middle age. Computers suffer from a similar problem – if not more accelerated. The current state of play here is that it is probably best to make sure that the people you buy the machine from deliver it unwrapped because, by saving a few moments in removing it from the packaging, you might just stand a chance of using it for a while before it is out of date. It is getting so bad that I have started looking for 'Best before .....' date stamps on the back of tower cases.

Of course when the animal gets to it's advanced age it does have a tendency to sit beside the fire staring with rheumy eyes into the near distance.

Fortunately for us our particular computational device is still very frisky. In spite of achieving a venerable old age there is a lot going on as the recent Paris show and a flurry of emails to the ql-users newsgroup have recently proved. When you look at this and then consider that I have just taken a vanload of 486 PCs, made after the QL ever emerged, to the local tip you can see that things are not that bad.

The Quanta Debate

Quanta's chairman, Robin Barker, has mooted the idea that the magazine goes 'multi-platform' end embraces other systems because its membership is sinking. (bit of a heavy pun there). I am not sure that this will solve many of the problems which exist here because, if the magazine goes in the PC arena, they are heavyweight PC magazines full of glossy pictures and swathes of advertising but it is a valiant effort to support the organisation which has supported the QL since its inception.

The ql-users newsgroup was buzzing with people willing both to attack and defend QUANTA recently but, whatever you think of the organization, it does support the people who arrange the shows and workshops and without these the whole community would begin to crumble even faster than it already has. You should also remember that QUANTA-took over production of the QIMI mouse when that ceased, reprinted the Jan Jones SuperBasic book and gained the publication rights to the Macro Assembler. It also provided two batches of Super Gold Cards when Miracle decided to stop their production so its support cannot be denied.

Some people took issue with the fact that you have to be a member to access the library and buy these items but, in the case of the library, these were the stipulations by which they obtained some of the stuff that they managed to get into it and, in the case of the hardware they would not have had the money to build it without the cash from the members.

Blinded by the Palette

The Paris show saw Tony Tebby seated before a Q 40 displaying a shimmering array of colours. The long awaited colour drivers were finally making an appearance on the scene. They is still in a very rudimentary form but much of the donkey work of coding the actual colours into the system has now been done.

We still have to get the colours into the programs and the SBASIC interface but there are now ways to produce 64K colours on the screen. This

threw up a number of odd effects, especially when applied to current programs. Where programmers had made a slight mistake in coding for a colour before the maximum range of 8 colours rarely produced anything unexpected. Now with all these various hues available we saw P. E. programs with green and yellow bands at the top instead of the more usual green and white.

Cueshell on my Q 40 is now configured to have a gorgeous blues surround and I have a copy of QD which almost psychedelic. Now I have your mouths watering for all of this I must point out that there are many programs that refuse to work in this version. ProWesS for one is smeared all over the page and several other programs which write directly to the screen are not at all happy. Some of this is due to holes in the current O/S because it is, as yet, unfinished and some of it will have to be cured by re-writing the programs themselves but we are most certainly getting close.

TT recently sent me a large _doc file with all of the specifications of the new drivers and with a cryptic message which suggests that he was very close to a beta version. I cannot reproduce any of this here because it was all marked confidential but my tongue is hanging out already.

Honourable Mentions in Dispatches

I have decided to add a new section to this column. From now on I will include the Honourable Mentions section to give a quick round of applause for worthy efforts in the previous two months. I hope this will stimulate you all to try to get a mention. I also hope that there will be enough new stuff around to make sure I can get this section in each issue.

Get the picture?

Not only do we have the improvements mentioned above but a new head has popped up above the parapet. Dave Westbury has produced a superb JPEG viewer. In this program you can view a selection of JPEGs on the screen just by putting them all in one subdirectory and then pointing the viewer at that directory. The pictures flash onto the screen in sequence, each one being displayed for a short period before moving on to the next. the program will only work on an Aurora or a Q 40 but the output is very good. With this program you can really see the difference between the display on the Q40 and the Aurora because the Q40 wins hands down. The Aurora just does not have the graphical muscle needed for these kind of high quality displays. The programming here is done without the use of the colour drivers and pokes the values directly to the hardware so the result is a bit dirty but, even so, it cause a few shock waves at our local user group when it was shown.

This is the kind of useful stuff that shows that the system is not quite down and out. As far as I know Dave has not really put any of his programs out before but, if this is anything to go by, we may see a some good stuff in the future.

Clip it.

The second of this months Honourable mentions goes to Duncan Neithercut. I discovered Duncans ClipScrapBaord on Tony Firshman's BBS while uploading and advert to him and I downloaded it to take a look. This is a very interesting piece of work, based on some other peoples previous program extensions. To quote the _txt file included with the program; 'ClipScrapBoard Concept This pointer driven program takes the Simon Goodwins DIY toolkit CLIP extension to Aurora and the Extended Environment. A modified version of the original clip code allows it to cope with the increased pixel width of Aurora Screens. Three programs are included. ClipScrapBoard_exe to clip text to the scrap, ScrapMC to access the Scrap and Pasteit_obj to paste text into an unlocked window.'

I used this to take a bit of text from one version of QD and drop it into this one. I could have done this with the facilities available in QD itself but the ClipSrcapBoard will clip from a whole range of programs and can be used to take screen shots and for all manner of uses. Again Duncan is not a regular contributor of QL programs but he responded swiftly when I pointed out a few problems and has since re-arranged the programs to make the package work very well indeed. There are two other programs in this collection. Pasteit_obj allows you paste text from the Scrap into another open window and ScrapMC_obj allows you to manipulate or view the current scrap contents.

I have been using Phil Jones' Scratch program for some time and find that very useful and some of the features are duplicated in this program but I find the two things to be complimentary and not exclusive and I welcome a new programmer onto the scene.

■

The QL Show Agenda

Here is an (incomplete) list of the QL Shows to which you can look forward in the first half of 2000. Some show dates are not confirmed yet, therefore we just list the month. When the exact date is given, then it is confirmed. We will keep you informed in the next issue of QL Today, which will be published way before the first show will be held.

19. Feb.	Eindhoven, The Netherlands St. Joris College, same venue as always.
27. Feb.	Hove, United Kingdom Excelsior Hotel, same venue as last year.
April	Quanta Annual General Meeting
13. May	Eindhoven, The Netherlands St. Joris College, same venue as always.
Spring	East Coast US Show. See also page 6 inside!

The Irish QL show has had to be postponed because of venue problems and is now hoped to happen in the Belfast area next June or July instead, to avoid clashes with the show calendar for the first quarter of 2000. Details from Darren Branagh at Q-Celt Computing.

Roy Brereton is liasing with overseas user groups on behalf of Quanta to arrange the QL2000 meeting. Roy can be contacted on email at roy@brereton94.freeserve.co.uk

The QL Today team wishes you a Merry Christmas and all the best for the New Year 2000.